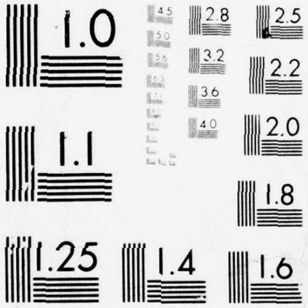
AD-A041 436   FRANK J SEILER RESEARCH LAB UNITED STATES AIR FORCE --ETC F/G 19/5
HIGH ANGLE OF ATTACK FLIGHT CONTROL USING STOCHASTIC MODEL REFE--ETC(U)
MAY 77 R B ASHER, D GOEBEL

UNCLASSIFIED   FJSRL-TR-77-0010   NL

1 OF 2
AD
A041436


1.0
4.5
5.0
5.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

2

12

FRANK J. SEILER RESEARCH LABORATORY

SRL-TR-77-0010 MAY 1977

HIGH ANGLE OF ATTACK FLIGHT CONTROL USING STOCHASTIC MODEL REFERENCE ADAPTIVE CONTROL

SCIENTIFIC

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

PROJECT 2304

AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

14

1. REPORT NUMBER
FJSRL-TR-77-Ø010 | AD-A

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

6

4. TITLE (*and Subtitle*)
HIGH ANGLE OF ATTACK FLIGHT CONTROL USING STOCHASTIC MODEL REFERENCE ADAPTIVE CONTROL

5. TYPE OF REPORT & PERIOD COVERED
Scientific

6. PERFORMING ORG. REPORT NUMBER

10

7. AUTHOR(s)
Robert B. Asher
Dieter Goebel

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Frank J. Seiler Research Laboratory (AFSC)
USAF Academy, Colorado 80840

16

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
DRS 61102F
2304-F1-54

17 F1

11. CONTROLLING OFFICE NAME AND ADDRESS
Frank J. Seiler Research Laboratory (AFSC)
USAF Academy, Colorado 80840

11

12. REPORT DATE
May 1977

13. NUMBER OF PAGES
105

14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)

12 1Ø6 P.

15. SECURITY CLASS. (*of this report*)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)
Model Reference Control
Adaptive Control
High Angle of Attack
High Angle of Attack Flight Control
Stochastic Control

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

High angle of attack flight control is of utmost importance to military aircraft in air combat maneuvering. Flight in this regime has in recent years caused many high performance aircraft to be lost due to departure of the aircraft. The aerodynamics in this regime are highly nonlinear. The problem is compounded by the fact that the aerodynamics are not well known. This paper considers the use of adaptive control in order to perform model following of an "ideal" aircraft in the presence

319 920

Abstract (cont)

of uncertain aerodynamic cuefficients. In particular, the partitioning approach of adaptive control is extended to the implicit model following problem. This is then used to solve the problem of high angle of attack flight control.

HIGH ANGLE OF ATTACK FLIGHT CONTROL USING STOCHASTIC MODEL REFERENCE ADAPTIVE CONTROL

by

[+]Robert B. Asher
Frank J. Seiler Research Laboratory
U.S. Air Force Academy, CO 80840

and

[++]Dieter Goebel
Wissenschaftlicher Direktor
BAKWVT, Mannheim, FRG

[+] Research Associate, Applied Mathematics Division

[++] Visiting Engineer, Frank J. Seiler Research Laboratory, when work was completed.

Table of Contents

List of Figures

Abstract

High angle of attack flight control is of utmost importance to military aircraft in air combat maneuvering. Flight in this regime has in recent years caused many high performance aircraft to be lost due to departure of the aircraft. The aerodynamics in this regime are highly nonlinear. The problem is compounded by the fact that the aerodynamics are not well known. This paper considers the use of adaptive control in order to perform model following of an "ideal" aircraft in the presence of uncertain aerodynamic coefficients. In particular, the partitioning approach of adaptive control is extended to the implicit model following problem. This is then used to solve the problem of high angle of attack flight control.

I. Introduction

High angle of attack flight may cause undesirable aircraft dynamic response. This dynamic response has been the cause for the loss of many high performance swept-wing aircraft due to stall-departure problems. Aircraft departure can be defined [1] as an uncommanded and/or uncontrollable dynamic response of the aircraft. It is manifested as either a divergent rolling-side slipping oscillation of large amplitude, i.e., wing rock, or a large rapid yaw generally followed by a rapid roll, i.e., nose slice. For example, reference [1] documents a simulation in which an A-7 airplane in a turn at high angle of attack flight exhibited nose slice with a yaw rate buildup of approximately 65 deg/sec. In an actual flight this would have caused a spin from which the probability of recovery would have been low. When this condition is encountered, it is unanticipated and the pilot is under physical and mental stress; the aircraft and pilot will most likely be lost if departure occurs during combat and certainly will be lost under 15,000 ft altitudes regardless of spin recovery characteristics [2].

The importance of departure led to a symposium at the Air Force Flight Dynamics Laboratory in 1971 on AFFDL Stall/Post Stall/Spin Symposium. Since then many articles have appeared in the study of departure [1,2,3,4,5]. Numerous studies have been made to identify the aerodynamic coefficients in high angle of attack regime. Reference [6] is a complete study of this.

However, in order to collect real time data for proper identification, it is necessary to subject the aircraft to divergence. Wind tunnel data, although valuable, still have residual errors due to differences in unsteady flow between the wind tunnel and the actual flight condition. It is desirable to accurately have knowledge as to the coefficients for control purposes; in fact, it is necessary.

This paper considers the control of aircraft at high angles of attack in the presence of uncertainty in the aerodynamic coefficients. In particular, the nonlinear equations of motion are linearized about a given flight condition. The nonlinear aerodynamics are included in the model. An ideal aircraft model for the given flight condition is developed. This ideal model varies with changes in flight condition. An adaptive implicit model following control law is developed in order to keep the actual aircraft close to the ideal response. Uncertainties in the aerodynamic coefficients are eliminated by the adaptive estimator.

This paper is divided into five sections. The next section contains the problem statement. Section III gives the development of the control law. Section IV contains a discussion as to how the ideal model for the A-7 was chosen. A different aircraft may have a different "ideal" model. Section V contains simulation results for the A-7 aircraft using the control law in the paper. Section VI yields the conclusions.

II. Problem Statement

The equations of motion of an aircraft linearized about a given angle of attack, α_o; Euler angles between gravity oriented inertial axis and aircraft body axis, θ_{po}, ℓ_o, ψ_o; the flight path angle, γ_o; angular velocities, p_o, q_o, and r_o; nominal forward velocity, U_o; sideslip angle, β_o, and given as

$$\dot{x} = A(\mu_1,t)x + B(\mu_2,t)u$$

where

$$x^T = \left\{u_v,\ \alpha-\alpha_o,\ q,\ \beta-\beta_o,\ p,\ r,\ \ell,\ \theta_p-\theta_{po}\right\}^T,$$

$$u^T = \left\{\delta_e,\ \delta_a,\ \delta_R\right\}^T$$

where u_v is the perturbed total linear velocity; $\alpha-\alpha_0$ is the perturbed angle of attack; p, q, and r are the perturbed angular velocities; $\beta-\beta_0$ is the perturbed sideslip angle; $\theta_p-\theta_{po}$ is the perturbed pitch angle and ϕ is the roll angle. The controls are assumed to be deflections in elevator, δ_e, in aileron, δ_a, and in rudder, δ_R. The matrices A and B are given in Appendix A as well as the definitions of μ_1 and μ_2. The parameter vector μ_1 and μ_2 contain the aerodynamic coefficients which are assumed uncertain except for an a priori probability density function. The coefficients are assumed constant over the time interval that the linearization is assumed valid. The time dependency of the A and B matrices depict the temporally changing linearization.

It is assumed that a noisy measurement of the states is available, i.e.,

$$y_m = Cx + v \tag{2}$$

where v is zero mean white noise with covariance $E\left\{v(t)v(\tau)^T\right\} = V_R\delta(t-\tau)$ and C is defined in Appendix A.

From Equation (A.2), it may be noted that the longitudinal and lateral modes of the aircraft are highly coupled. Furthermore, as may be noted by expansion of the equations of motion, there are many destabilizing terms in the equations. Consequently, it is an extremely difficult multivariable task for the pilot to prevent departure at high angles of attack. This is especially true in air combat maneuvering as the physical and mental stress occupies the pilot's attention. Thus, it is desirable to obtain a closed loop control law which will prevent departure. One method is reported in [5] where the author develops a feedback control to eliminate perturbations about a nominal trajectory for a deterministic system. The approach in this paper is to design a feedback control law based on model following of an ideally responding aircraft. The advantage is that the model following control more closely gives a control law that will yield a desirable response. Thus, for example, decoupling of the yaw-roll problem in nose slice may be approximately obtained by placing this feature into the ideal model. The destabilizing terms may be compensated by use of the model. The adaptive control law will compensate for uncertainties in the aerodynamic coefficients by real time learning. The development of the ideal model will be explained in Section IV with a specific example given for the A-7 in a particular flight condition. The form of the model is

$$\dot{z} = A_m(t)z \tag{3}$$

where the time varying A_m matrix corresponds to the ideal model changing due to the different flight conditions. Since implicit model following [10] is to be accomplished, the performance index is taken to be

$$J = E\left\{\int_{t_o}^{t_f}\left[(\dot{y}_o - A_m y_o)^T Q_p(\dot{y}_o - A_m y_o) + u^T R_p u\right]dt\right\} \tag{4}$$

where y_o is the output vector (not to be confused with the measurement vector)

$$y_o = C_o x \tag{5}$$

where C_o is a time invariant distribution matrix and y_o is a general term since x and z may not be of the same dimensions and where R_p weights the control surface deflections and is a positive definite matrix, Q_p weights excursions from the model response, and t_f is chosen as the interval over which the linearization and constancy of the aerodynamic coefficients are assumed valid. The optimal control would fall into the class of dual control problems. However, additional uncertainties in the model other than those accounted for, unsteady flow problems, as well as the survivability dictates that the dual control may not be used. This is because additional control responses due to the identification aspect of dual control may, because of additional uncertainties, cause an extremely undesirable response leading, perhaps, to an aggravation of the divergence problem. This may in an extreme case lead to the loss of the aircraft. Thus, an adaptive open loop feedback controller will be chosen. There are two major techniques that may be used. The first in reference [7] yields the optimal open loop feedback controller for the problem with uncertain parameters. The computational burden of the technique is larger than the second technique of [8] even though it will lead to better performance as it is an optimal technique. Since optimality certainly must, in this application, include computational burden, the technique as given in [8] will be extended to the model following problem.

The partitioning technique as in [8] consists of using a control law found by solving the μ-conditional control problem and then weighting the μ-conditional control with the probability density of μ conditioned on the measurements. The technique includes the measurement conditional probability density function for μ in tne solution for the control gains. This yields the optimal open-loop feedback control solution, but it has the disadvantage that the equations for the control gain differ at each measurement. In this paper a control law with as little computational burden as possible consistent with the uncertainty problem and good performance is desired. Consequently, the partitioning algorithm will be chosen as the adaptive control method in this paper.

III. Control Law Development

The solution to the partitioned adaptive control model reference problem may be found by using state augmentation techniques. The new state ζ is defined as

$$\zeta^T = [x^T, z^T].$$

This yields a state equation as

$$\dot{\zeta} = \overline{A}(\mu_1)\zeta + \overline{B}(\mu_2)u \tag{6}$$

where

$$\overline{A}(\mu_1) = \begin{bmatrix} A(\mu_1) & 0 \\ 0 & A_m \end{bmatrix}$$

and

$$\overline{B}(\mu_2) = \begin{bmatrix} B(\mu_2) \\ 0 \end{bmatrix}.$$

The performance index may be easily rewritten as

$$J = E\left\{\int_{t_o}^{t_f}\left[x^T\overline{Q}(\mu_1)x + 2u^Ts(\mu_1,\mu_2)x + u^T\overline{R}(\mu_2)u\right]dt\right\} \tag{7}$$

where

$$\begin{aligned}\overline{Q}(\mu_1) &= \left[C_oA(\mu_1) - A_mC_o\right]^TQ_p\left[C_oA(\mu_1) - A_mC_o\right],\\ S(\mu_1,\mu_2) &= B(\mu_2)^TC_o{}^TQ_p\left[C_oA(\mu_1) - A_mC_o\right], \qquad (8)\end{aligned}$$

and

$$\overline{R}(\mu_2) = B(\mu_2)^TC_o{}^TQ_pC_oB(\mu_2) + R_p.$$

Thus, it may be noted that the integral under the performance index as well as the system dynamics are functions of μ_1 and μ_2.

The partitioned adaptive control law may be found by solving for the deterministic control gain conditioned on μ_1 and μ_2, for the μ_1, μ_2 conditional Kalman filter estimate, $\hat{x}(t|\mu_1,\mu_2,\psi_t)$, and for the conditional density, $p(\mu_1,\mu_2|Y_t)$ where $Y_t = \{y(\tau),t_o \leq \tau \leq t\}$, and using as a control

$$u(t) = \int_{R_{\theta_1}}\int_{R_{\theta_2}} \overline{K}(t|\mu_1,\mu_2)\hat{x}(t|\mu_1,\mu_2,\psi_t)p(\mu_1,\mu_2|Y_t)d\mu_1d\mu_2. \tag{9}$$

If μ_1 and μ_2 are defined over discrete ranges, then equation (9) may be rewritten as

$$\begin{aligned}u(t) = \sum_{i=1}^{\ell_1}\sum_{j=1}^{\ell_2} &\overline{K}(t|\mu_{1_i},\mu_{2_j})\hat{x}(t|\mu_{1_i},\mu_{2_j},Y_t)\\ &\cdot P_r(\mu_{1_i},\mu_{2_j}|Y_t)\end{aligned} \tag{10}$$

where $P_r((\cdot))$ denotes the probability of the event $(\cdot)$. The control gain may be determined by the solution of the μ_{1_i} and μ_{2_j} conditional deterministic problem, i.e.,

$$\overline{K}(t|\mu_{1_i},\mu_{2_j}) = -\overline{R}(\mu_{2_j})^{-1}[S(\mu_{1_i},\mu_{2_j}) + B(\mu_{2_j})^T P(\mu_{1_i},\mu_{2_j},t)] \tag{11}$$

where

$$\begin{aligned}
\dot{P}(\mu_{1_i},\mu_{2_j},t) = {} & P(\mu_{1_i},\mu_{2_j},t)[A(\mu_{1_i}) - B(\mu_{2_j})\overline{R}(\mu_{2_j})^{-1} \\
& \cdot S(\mu_{1_i},\mu_{2_j})] + [A(\mu_{1_i}) \\
& - B(\mu_{2_j})\overline{R}(\mu_{2_j})^{-1}S(\mu_{1_i},\mu_{2_j})]^T P(\mu_{1_i},\mu_{2_j},t) \\
& - P(\mu_{1_i},\mu_{2_j},t)B(\mu_{2_j})\overline{R}(\mu_{2_j})^{-1}B(\mu_{2_j})^T P(\mu_{1_i},\mu_{2_j},t) \\
& + \overline{Q}(\mu_{1_i}) - S(\mu_{1_i},\mu_{2_j})^T\overline{R}(\mu_{2_j})^{-1}S(\mu_{1_i},\mu_{2_j})
\end{aligned} \tag{12}$$

$$V_i = 1,2,\ldots,\ell_1$$
$$V_j = 1,2,\ldots,\ell_2$$

with final condition

$$P(\mu_{1_i},\mu_{2_j},t_f) = 0, \quad Vi,j.$$

The filter equations are the standard Kalman equations

$$\begin{aligned}
\dot{\hat{x}}(t|\mu_{1_i},\mu_{2_j},Y_t) = {} & A(\mu_{1_i})\hat{x}(t|\mu_{1_i},\mu_{2_j},Y_t) \\
& + B(\mu_{2_j})\overline{K}(t|\mu_{1_i},\mu_{2_j})\hat{x}(t|\mu_{1_i},\mu_{2_j},Y_t) \\
& + K_G(t|\mu_{1_i},\mu_{2_j})[y_m - C\hat{x}(t|\mu_{1_i},\mu_{2_j},Y_t)]
\end{aligned} \tag{13}$$

with

$$\hat{x}(t_o|\mu_{1_i},\mu_{2_j}) = \hat{x}(t_o)$$

where

$$K_G(t|\mu_{1_i}) = V(t|\mu_{1_i})C^T V_R^{-1} \tag{14}$$

with

$$\dot{V}(t|\mu_{1_i}) = A(\mu_{1_i})V(t|\mu_{1_i}) + V(t|\mu_{1_i})A(\mu_{1_i})^T - V(t|\mu_{1_i})C^T V_R^{-1} CV(t|\mu_{1_i}) \tag{15}$$

$$i = 1,2,\ldots,\ell_1$$

$$j = 1,2,\ldots,\ell_2$$

and

$$V(t_o|\mu_{1_i}) = V(t_o).$$

The conditional probability density function for μ_{1_i} and μ_{2_j} may be computed via

$$Pr(\mu_{1_i},\mu_{2_j},Y_t) = \frac{P(\mu_{1_i})P(\mu_{2_j})\exp\left\{\int_{t_o}^{t} x^T(t|\mu_{1_i},\mu_{2_j},Y_t)C^T V_R^{-1} y_m(t)dt - \int_{t_o}^{t} ||C\hat{x}(t|\mu_{1_i},\mu_{2_j},Y_t)||^2_{V_R^{-1}} dt\right\}}{\sum_{i=1}^{\ell_1}\sum_{j=1}^{\ell_2} P(\mu_{1_i})P(\mu_{2_j})\exp\left\{\int_{t_o}^{t} \hat{x}(t|\mu_{1_i},\mu_{2_j},Y_t)C^T V_R^{-1} y(t_o)dt - \int_{t_o}^{t} ||C\hat{x}(t|\mu_{1_i},\mu_{2_j},Y_t)||^2_{V_R^{-1}} dt\right\}} \tag{16}$$

$$i = 1,2,\ldots,\ell_1$$

$$j = 1,2,\ldots,\ell_2$$

where $P(\mu_{1_i})$ and $P(\mu_{2_j})$ are the a priori probabilities for μ_{1_i} and μ_{2_j}, respectively. The simulations use the proper discrete form of the equation as in reference [9].

The procedure is, thus, to first obtain a set of linearized equations where the linearization is taken over the current flight condition. This linearization must be updated frequently as the changing flight regime may drastically change the dynamic response. It is assumed that over a given time period the aerodynamic coefficients are constant. This assumption is valid over the same region that the linearization assumption is valid. However, the aerodynamic coefficients are not exactly known. It is assumed that the uncertain coefficients are defined over a discrete range. This procedure defines the linear equations as in equation (1). The ideal model which is dependent on the angle of attack and sideslip angle is utilized along with the measurement equation as in (2) along with the determination of the output equation (5) in order to define the remaining equations for dynamic response. The control weighting matrix R_p and model matching weighting matrix Q_p must be determined. The final time, t_f, must be determined. This is the maximum time that the linearization will be assumed valid.

Thus, all the equations necessary for control law design are now assumed to be available. Equations (13-15) are used to obtain the μ_1 and μ_2 conditional state estimates for the aircraft state. This is used in equation (16) to find the probability of each μ_1 and μ_2. The control gain as in (11) is calculated, and the control is determined from equation (10).

An approximate law may be obtained by finding the steady state gains $\overline{K}$ and the steady state Kalman filter and using these in the control computations.

IV. Ideal Aircraft Model

This section uses the equations for the A-7 as given in Appendix A in order to discuss the reasons for departure of the A-7 and in order to yield an ideal model with better response in the high angle of attack regime.

The equations of motion for the actual A-7 are given in Appedix A. They are formulated using wind axes and the aerodynamic derivatives are evaluated at a prestall flight condition of $\alpha_0 = 19$ deg and $\beta_0 = 6$ deg. The incremental change in rolling moment L_i' and in yaw moment N_i' are calculated by

$$L_i' = \frac{L_i + (I_{xz}/I_x)N_i}{1 - (I_{xz}^2/I_xI_y)}$$

and (17)

$$N_i' = \frac{N_i + (I_{xz}/I_x)L_i}{1 - (I_{xz}^2/I_xI_y)}$$

where i denotes the particular state variable and where I_x and I_y are body axes moments of inertia, I_{xz} is the respective cross product of inertia, and L_i and N_i are the aerodynamic moments about the conventional aircraft body axes. As mentioned in reference [1], the major coupling which affects departure for the A-7 is provided by the kinematic terms

$$Z_p = \beta_o \cos \alpha_o$$

and (18)

$$Z_r = \beta_o \sin \alpha_o$$

(see equation A.2), and the aerodynamic terms L'_α and N'_α. The kinematic terms arise due to the rotating coordinate system. The aerodynamic terms L'_α and N'_α as well as L'_β and N'_β are given as functions of α and β in Figure 9 of reference [1]. In the regime of α_{stall} (the stall angle of attack) these aerodynamic terms change sign and, therefore, an aerodynamic term stabilizing at small α can contribute to the tendency for departure at high α.

The A-7 has a typical nose slice departure. The influence of the main aerodynamic derivatives on this departure is discussed at length in reference [1]. Therefore, in the following, only points pertinent to finding the ideal model are discussed. This discussion is based upon many simulation runs as well as physical insight.

The influence of the "effective derivatives"

$$Z_p = \beta_o \cos \alpha_o$$

and (19)

$$Z_r = \beta_o \sin \alpha_o$$

is not a major influence on the ideal model. Consequently, the ideal model contains these terms in their original form. In some of the simulations they were zeroed and the results were not changed significantly.

For static lateral stability in yaw, N'_β should be positive. Figure 9 in reference [1] shows that a negative N'_β can be expected for an angle of attack greater than 17 deg. For a flight with high angle of attack and small sideslip, $N'_\beta < 0$ will increase the sideslip angle and will, therefore, increase N'_α which is, in the high angle of attack, always destabilizing. The result is the high yaw rate of the aircraft.

For static lateral stability in roll, L'_β should be negative. This means that a positive unwanted bank angle (right wing down) will induce a positive sideslip which causes a negative rolling moment with a decrease in bank angle as a result. A negative L'_β and a negative L'_α are the primary reasons for departure of the A-7 after a rapid yaw. The L'_α is negative if $\alpha > 23$ deg and the magnitude increases with sideslip.

The desired model for implicit model following was obtained with several goals in mind. The terms in the state model were chosen with the following results. The yaw moment due to sideslip should be positive, $N'_\beta > 0$. This will decrease the sideslip and therefore the destabilizing influence of L'_α and N'_α. The roll moment due to sideslip should be negative, $L'_\beta < 0$. An artificial static roll stability ($L_\theta < 0$) was introduced instead of the "natural" static roll stability ($L'_\beta < 0$) which can contribute to departure.

The model was chosen with the above criteria satisfied. Simulations were conducted to help choose and to verify the model picked. These were compared to the actual A-7 in the same flight regime with significant improvement. The model matrix chosen for the flight condition about $\alpha_0 = 19^0$ and $\beta_0 = 6^0$ is

$$A_m = \begin{bmatrix} -0.0634 & -22.68 & 0 & -5.766 & 0 & 0 & 3.187 & -32.024 \\ -0.0009 & -0.323 & 1.0 & 0 & -0.0995 & -0.0338 & 0 & 0 \\ 0 & -3.577 & -0.386 & 0 & -0.0082 & 0.0025 & 0 & 0 \\ 0 & 0.0122 & 0 & -0.1062 & 0.3216 & -0.9469 & 0.1166 & 0.0129 \\ 0 & 0 & 0 & -1.0 & -0.849 & 0.3323 & -0.5 & 0 \\ 0 & 0 & 0 & 1.5 & 0.0193 & -0.1276 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0.3397 & 0 & -0.0116 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0.0104 & 0 \end{bmatrix} \quad (20)$$

The available controls are aileron deflection, elevator deflection, and rudder deflection. The thrust is constant over the flight.

The next section yields the results.

V. Simulation Results

The control law was used to find the adaptive implicit model following law for the A-7 about the $\alpha_0 = 19^0$ and $\beta_0 = 6^0$ case. The model used is defined in equation (20). Measurements of all the states corrupted by white noise were assumed available. The standard deviations for the measurement noise are as follows: velocity perturbation, 1.25 ft/sec; angle of attack perturbation, 0.005 rads; pitch rate, 0.01 rad/sec; sideslip perturbation, 0.005 rad; roll rate, 0.01 rad/sec; yaw rate, 0.01 rad/sec; roll angle, 0.005 rad; and pitch angle perturbation, 0.005 rad. In each figure for this case, the nomenclature A-7 corresponds to an open loop ($\delta_e = \delta_R = \delta_a = 0$) simulation of the A-7 with an initial yaw rate of -10 deg/sec. As is shown in reference [1] a control input typical of an actual pilot response does not control the departure. The nomenclature MO corresponds to an ideal model which the control is calculated to follow, and the nomenclature MF corresponds to the actual A-7 response with the closed loop control calculated in this paper. Equal control weighting was used.

It is assumed that the coefficients L'_α, N'_α and L'_p are uncertain. These coefficients have a major impact on lateral directional stability. The true value of the parameters were 3.09, -1.486, and -0.849, respectively. It was assumed that the possible parameter vectors, $\{L'_\alpha, N'_\alpha, L'_p\}^T$, were 1.9, 1.45, 1.0, 0.55 and 0.1 times the true values. That is, the possible parameter set was contained within a set of five possible values. The adaptation took place on these three aerodynamic coefficients.

Figures 1-8 show the radical difference in response using the control law derived in this paper. The actual A-7 shows a buildup of roll rate (Figure 5) followed by a rapid increase in bank angle (Figure 7). This type of behavior can indeed cause the loss of the aircraft. The responses using the closed loop control show that divergence is prevented. The response is very adequate using this control. Figures 9-11 show the control deflections required for divergence prevention for this lateral directional case.

Figure 12 shows the probabilities of each parameter being the true parameter. It takes less than 1.75 seconds to adapt upon the correct parameter with probability 0.8.

Case two was chosen to show the coupling between longitudinal and lateral dynamics. This case starts with a 5 deg/sec pitch rate initial condition (Figure 13). Figures 14-16 show a buildup in the A-7 response in the lateral modes due to the initial longitudinal pitch rate initial condition at the high angle of attack regime. Figure 17 vividly depicts the buildup of bank angle as the aircraft goes into departure without the control law used. These figures also show that with the control law applied with deflections in Figures 18-20 that the aircraft is prevented from departure. The control laws in essence yield a soft decoupling of modes while controlling the aircraft.

Figures 21-28 show the standard deviations of the estimation error for each of the conditional Kalman filters. The true parameter is number 3 in these figures.

Several additional simulations were conducted with different flight conditions as well as noise sequences. Each result is very similar to these typical results.

VI. Conclusions

The control law and philosophy of flight control developed within is shown to be an excellent method of divergence prevention in the high angle of attack regime. The control laws found by finding steady state gains for the filters as well as the control gains may be readily implemented along with the probability estimator for uncertain coefficients. The control law was simulated in detail and shows excellent promise for control in a dangerous flight regime.

The model development philosophy points out many key problems in the high angle of attack regime. This information of itself is valuable.

References

1. D.E. Johnston and J.R. Hogge, "Nonsymmetric Flight Influence on High Angle of Attack Handling and Departure," AIAA Journal of Aircraft, Vol. 13, No. 2, pp. 112-118, February 1976.

2. W. Bihrle, Jr. and R.C. Meyer, "F-14A High Angle of Attack Characteristics," AIAA Journal of Aircraft, Vol. 13, No. 8, pp. 576-583, August 1976.

3. R. Weissman, "Preliminary Criteria for Predicting Departure Characteristics/ Spin Susceptibility of Fighter-Type Aircraft," AIAA Journal of Aircraft, Vol. 10, No. 4, pp. 214-219, April 1973.

4. R.F. Stengel and P.W. Berry, "Stability and Control of Maneuvering High Performance Aircraft," Proceedings of the 1976 AIAA Guidance and Control Conference, AIAA Paper 76-1973, pp. 485-492.

5. K.L. Johnson and T.B. Willen, "Anticipated Spin Susceptibility Characteristics of the A-10 Aircraft," AIAA Journal of Aircraft, Vol. 13, No. 6, pp. 406-412, June 1976.

6. W.E. Hall, Jr., N.K. Gupta, and R.G. Smith, "Identification of Aircraft Stability and Control Coefficients for the High Angle of Attack Regime," Systems Control, Inc., Technical Report #2, March 1974.

7. R.B. Asher, C.S. Sims, and H.R. Sebesta, "Optimal Open-Loop Feedback Control for Linear Systems with Unknown Parameters," Information Sciences, Vol. 11, No. 3, pp. 265-277, 1976.

8. D.G. Lainiotis, "Partitioning: A Unifying Framework for Adaptive Systems, II: Control," Proceedings of the IEEE, Vol. 64, No. 8, pp. 1182-1198, August 1976.

9. D.G. Lainiotis, "Optimal Adaptive Estimation: Structure and Parameter Adaptation," IEEE Transactions on Automatic Control, Vol. AC-16, No. 2, pp. 160-170, April 1971.

10. E. Kreindler and D. Rothschild, "Model Following in Linear Quadratic Optimization," AIAA Journal, Vol. 14, No. 7, pp. 835-842, July 1976.

11. D.E. Johnston, I.L. Ashkenas, and J.R. Hogge, "Investigation of Flying Qualities of Military Aircraft at High Angles of Attack, Vol. I: Technical Results," AFFDL-TR-74-61, June 1974.

12. D.E. Johnston, J.R Hogge, and G.L. Teper, "Investigation of Flying Qualities of Military Aircraft at High Angles of Attack, Vol. II: Appendices," AFFDL-TR-74-61, June 1974.

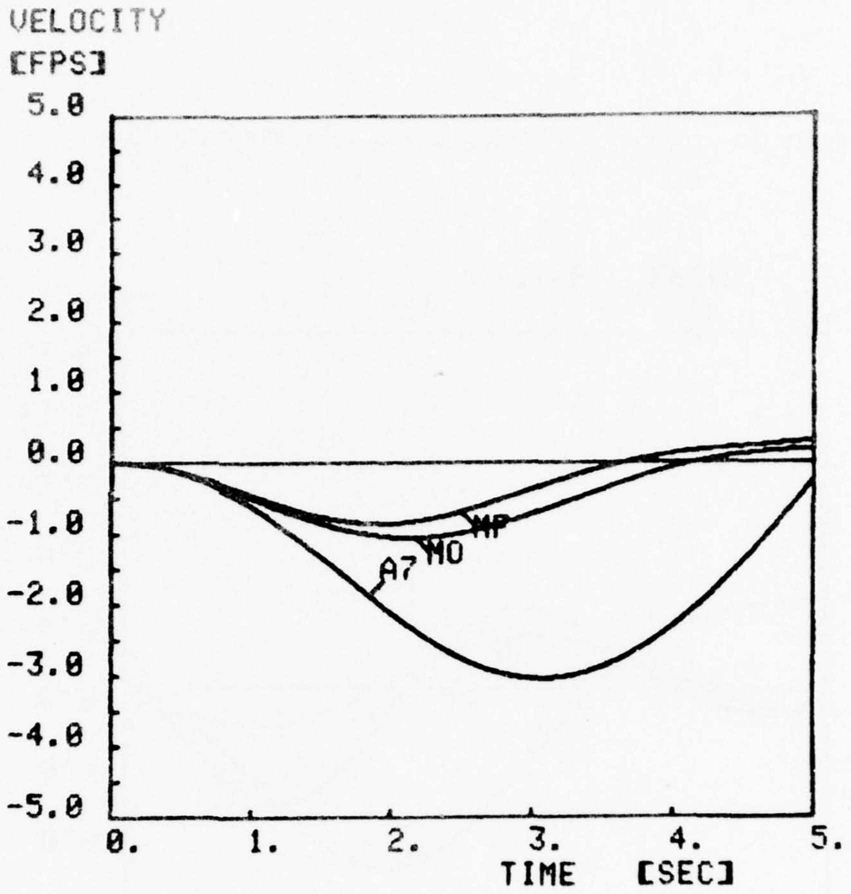


Fig. 1. Case 1 Aircraft Responses in Velocity

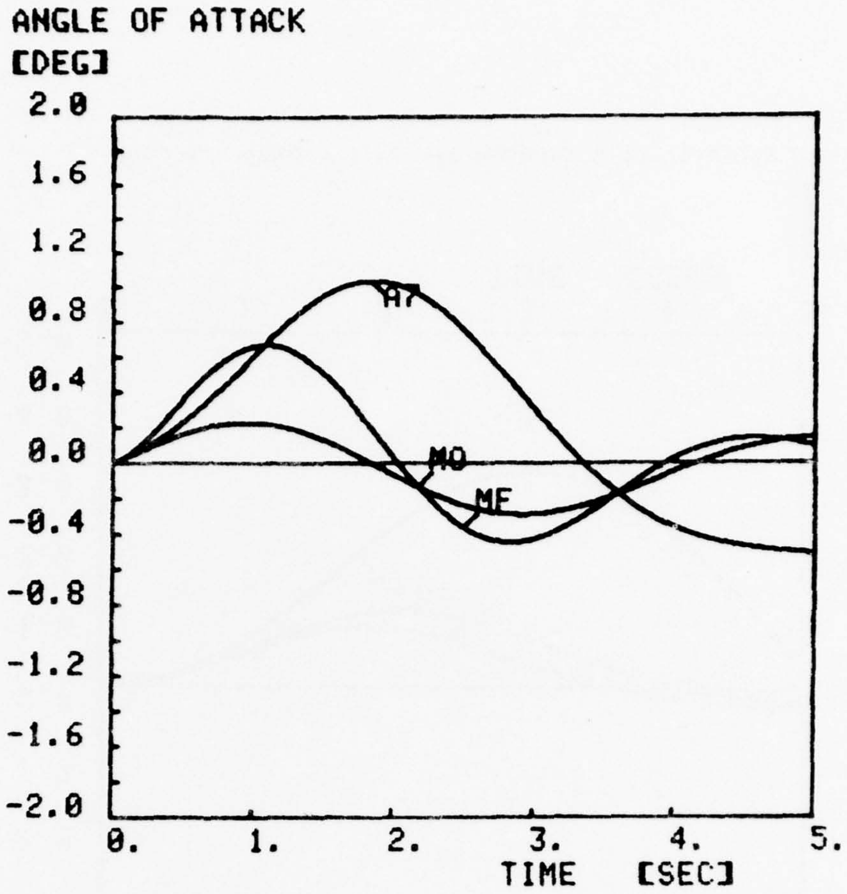


Fig. 2. Case 1 Aircraft Responses in Angle of Attack

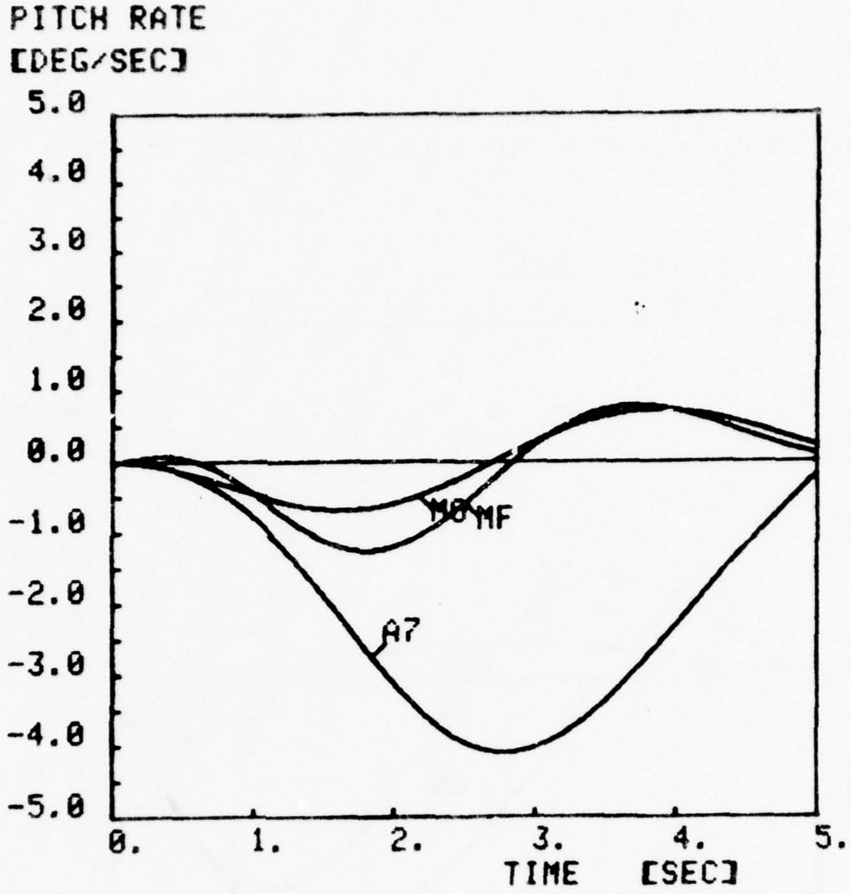


Fig. 3. Case 1 Aircraft Responses in Pitch Rate

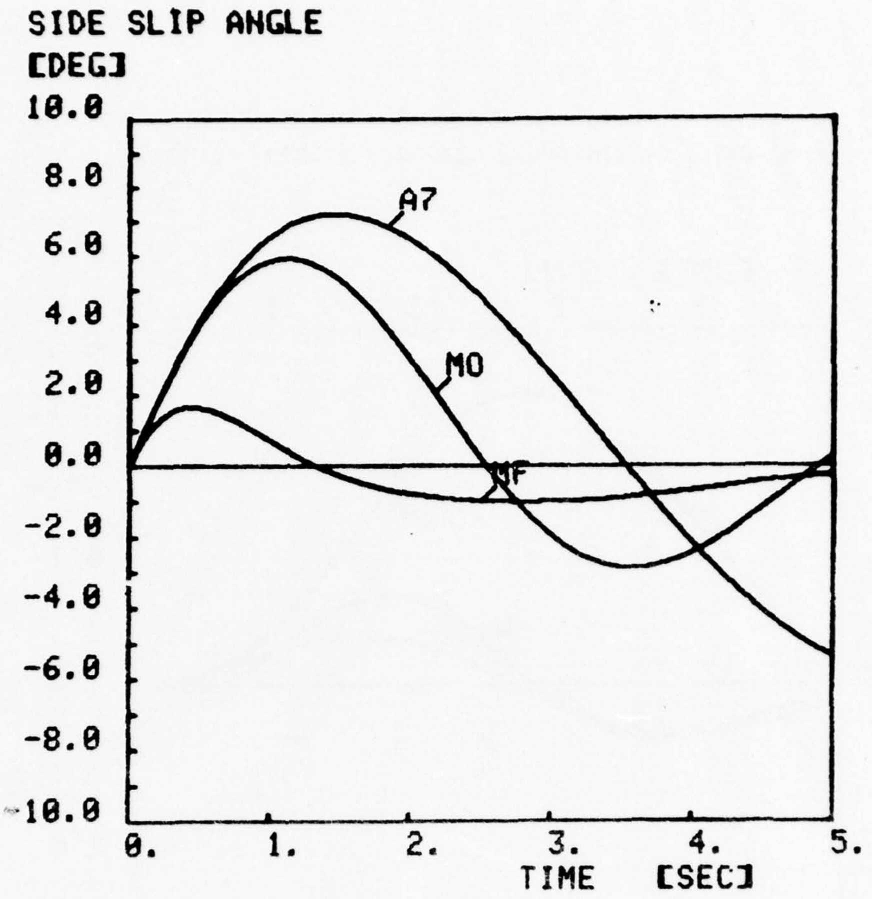


Fig. 4. Case 1 Aircraft Responses in Sideslip Angle

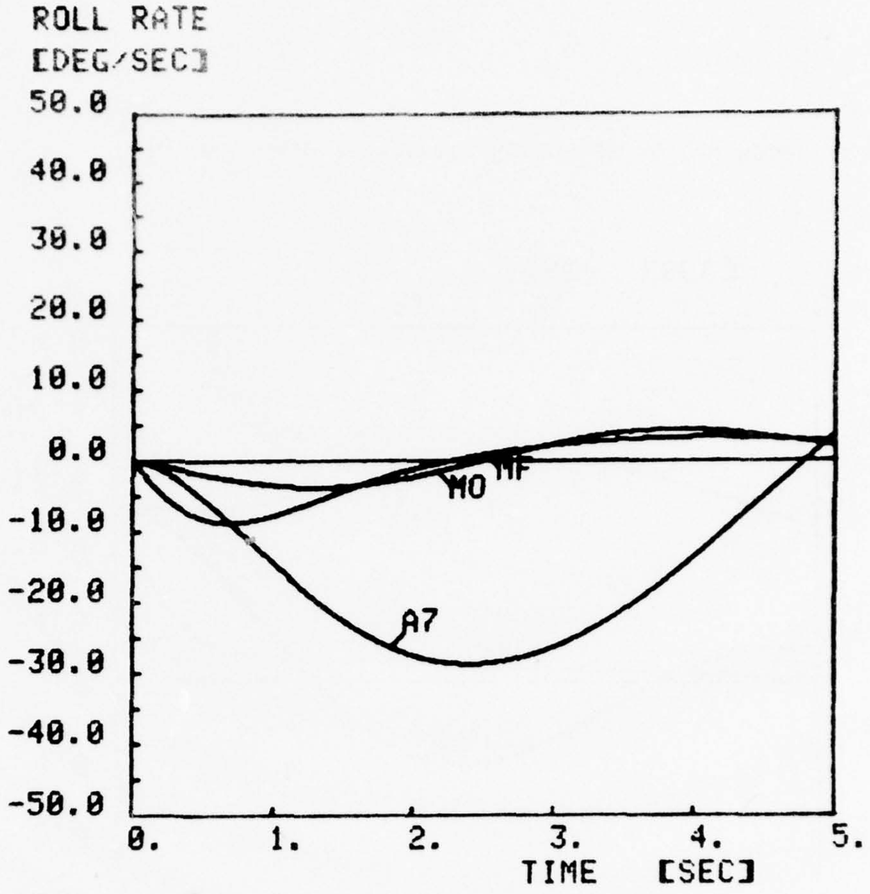


Fig. 5. Case 1 Aircraft Responses in Roll Rate

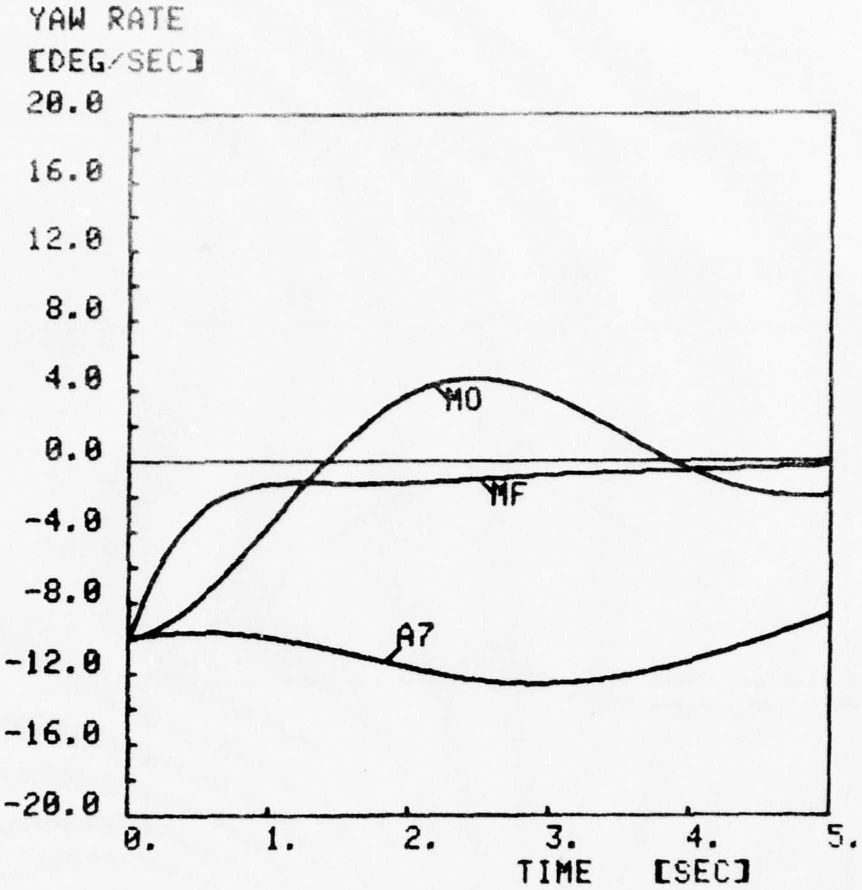


Fig. 6. Case 1 Aircraft Responses in Yaw Rate

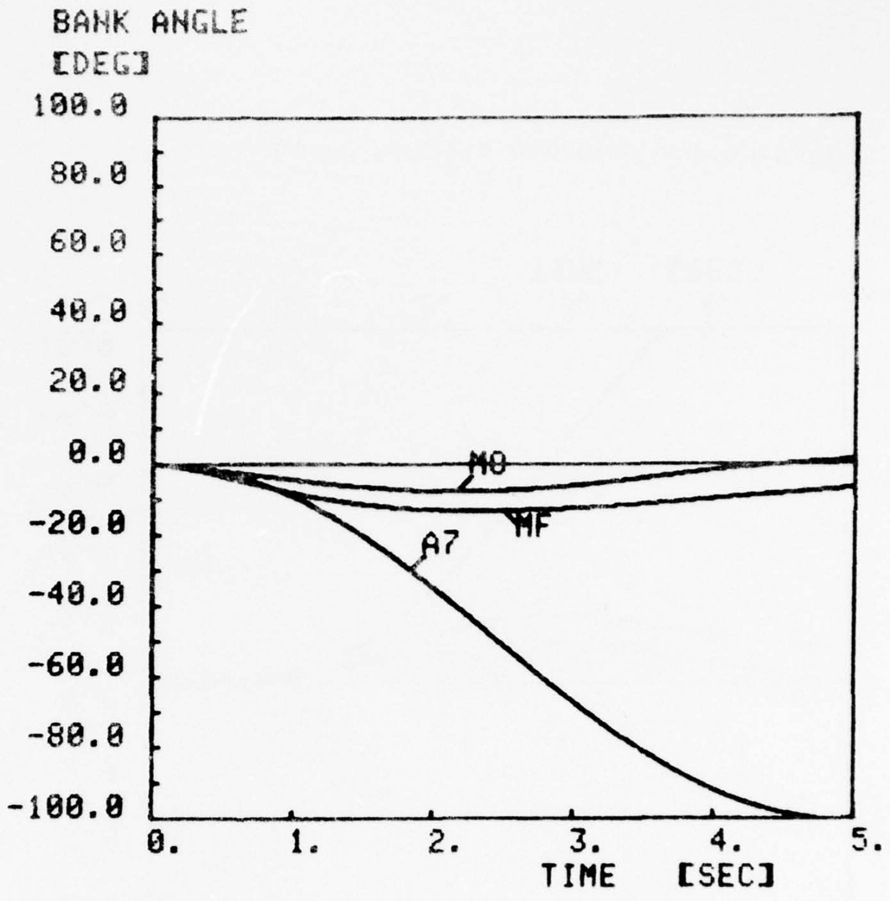


Fig. 7. Case 1 Aircraft Responses in Bank Angle

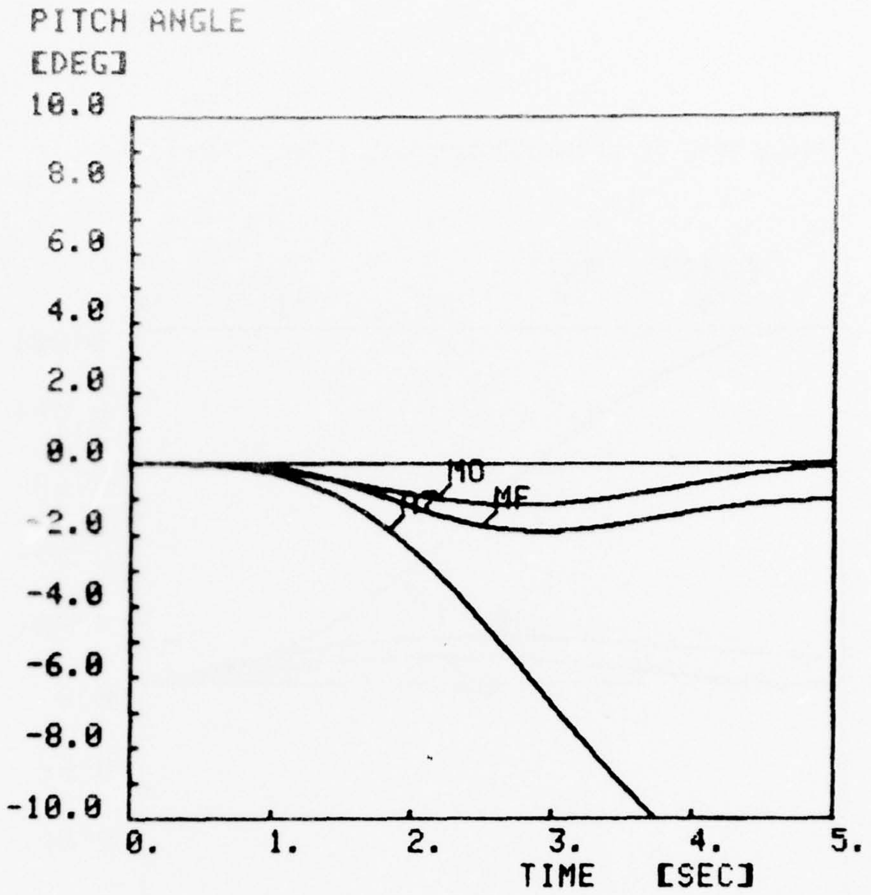


Fig. 8. Case 1 Aircraft Responses in Pitch Angle

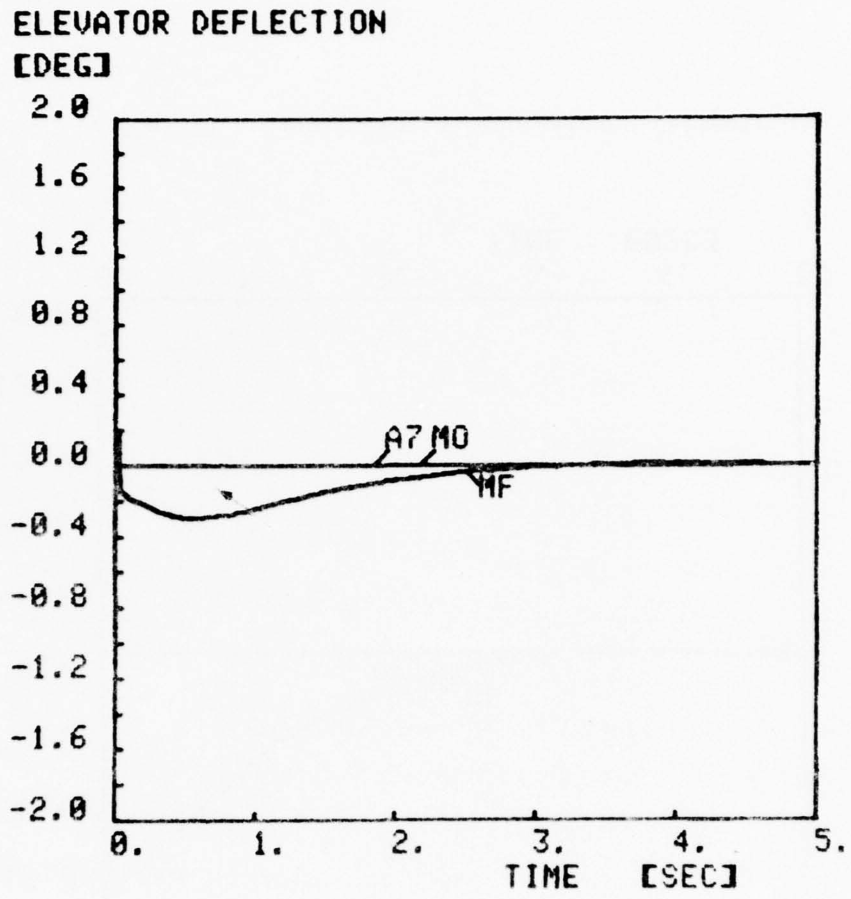


Fig. 9. Case 1 Closed Loop Elevator Deflection

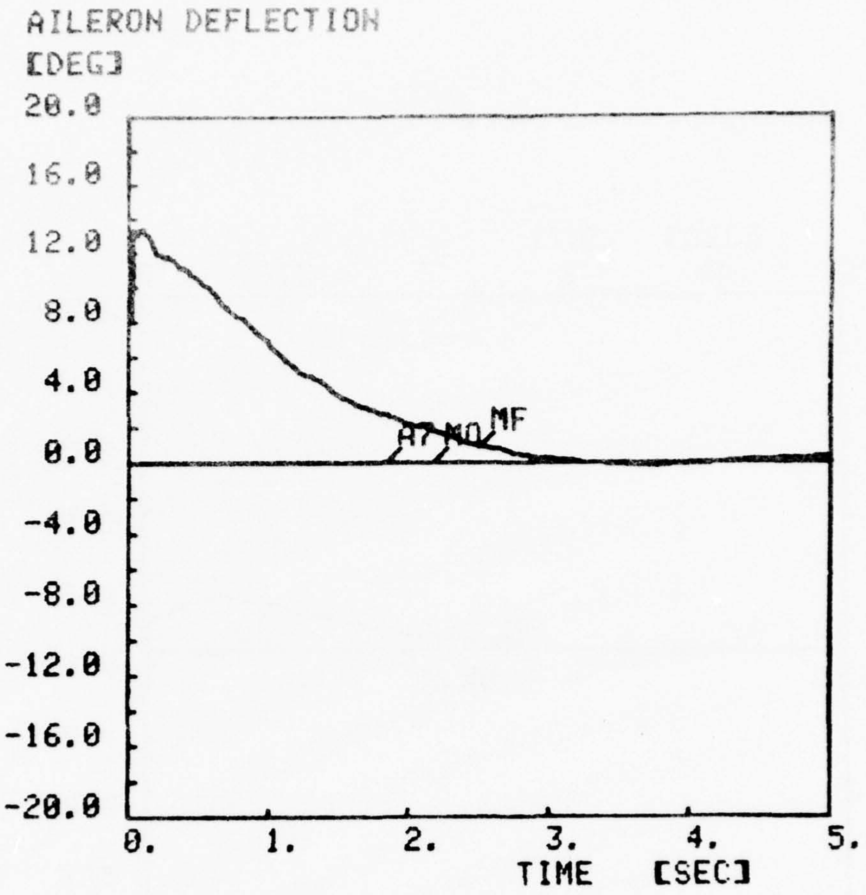


Fig. 10. Case 1 Closed Loop Aileron Deflection

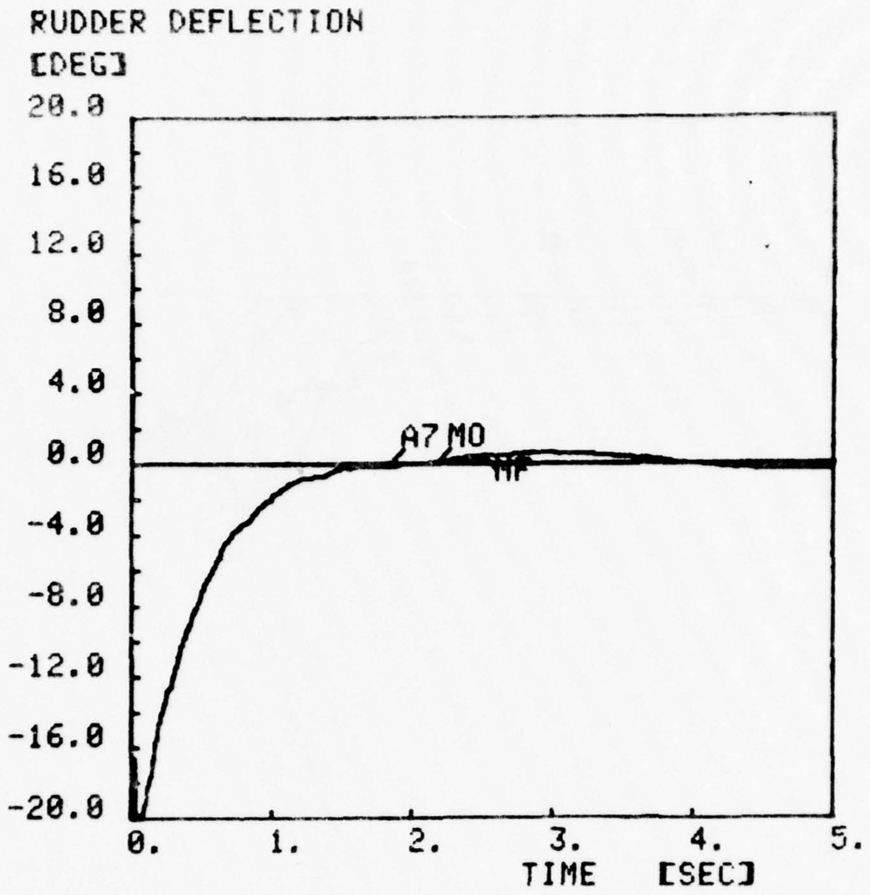


Fig. 11. Case 1 Closed Loop Rudder Deflection

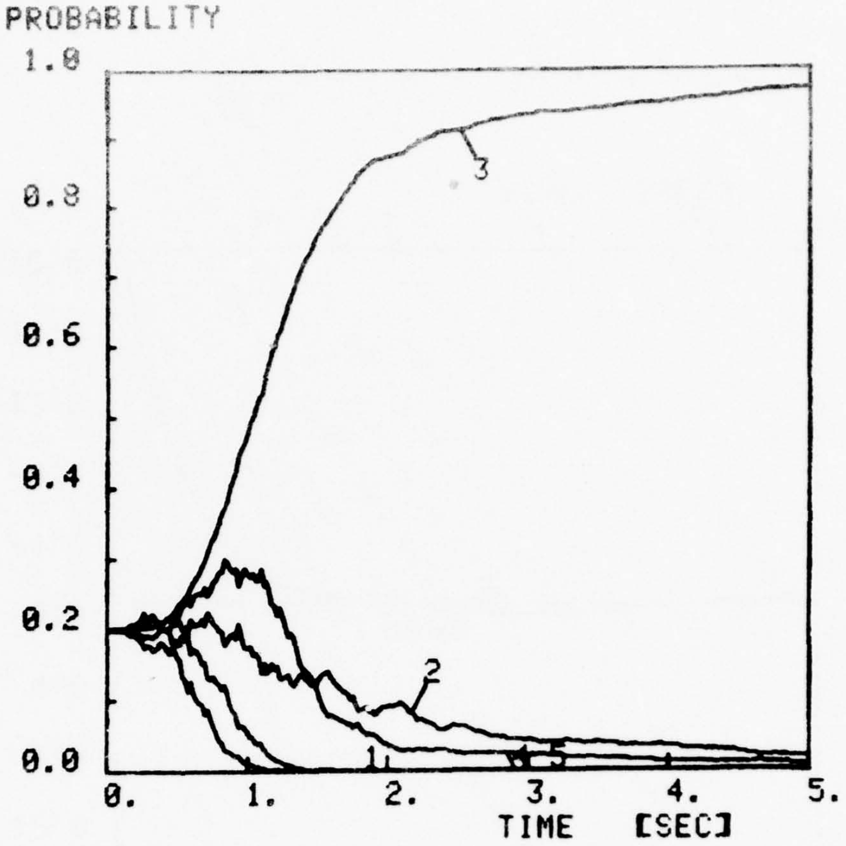


Fig. 12. Probabilities vs Time

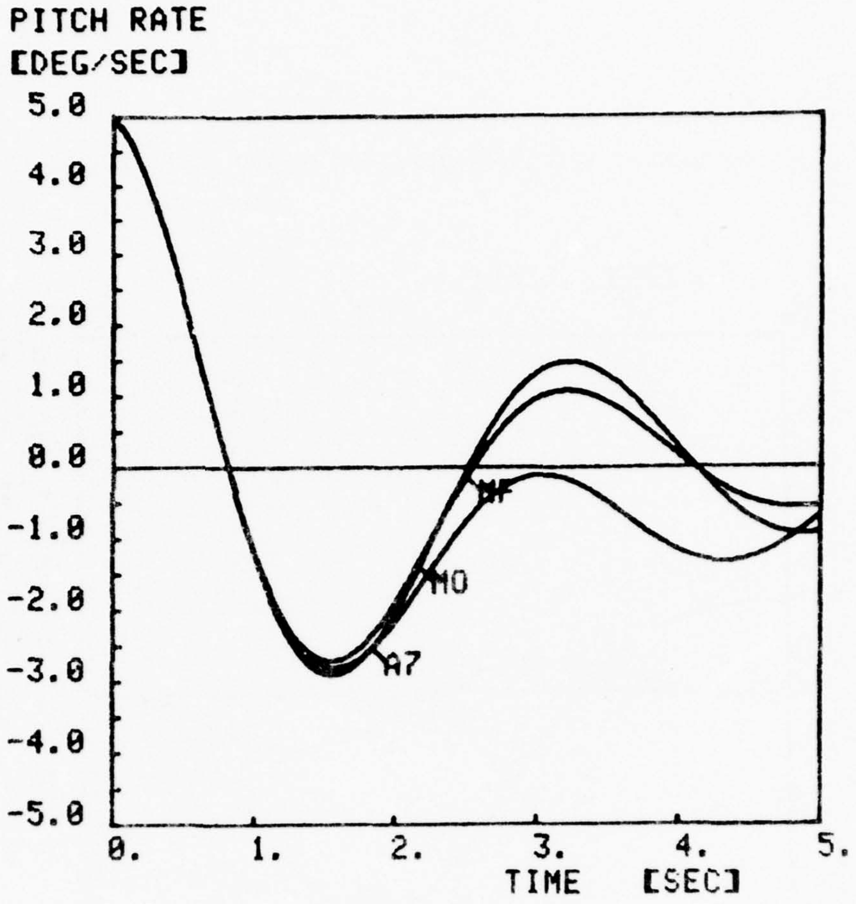


Fig. 13. Case 2 Aircraft Responses in Pitch Rate

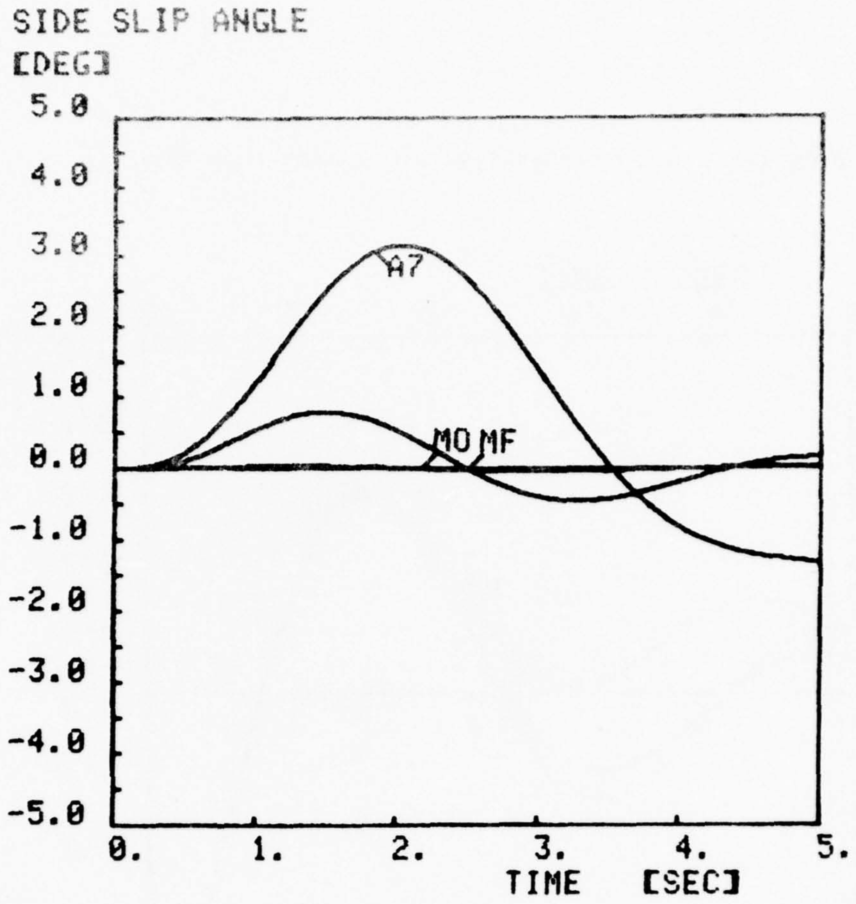


Fig. 14. Case 2 Aircraft Responses in Sideslip

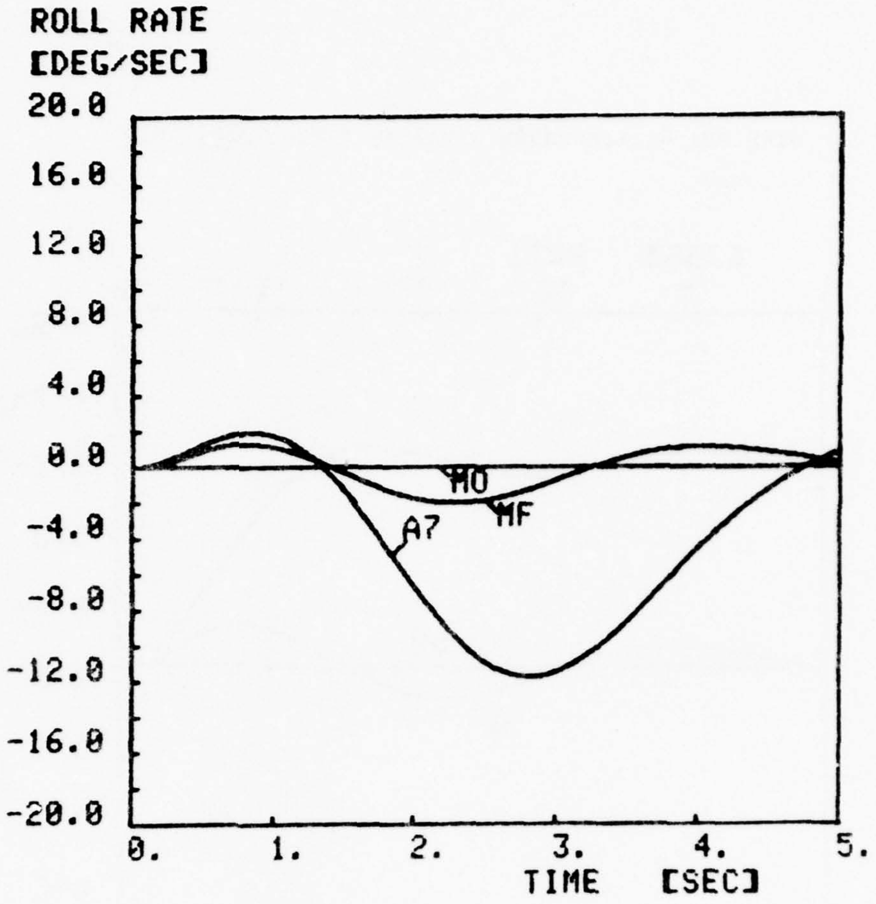


Fig. 15. Case 2 Aircraft Responses in Roll Rate

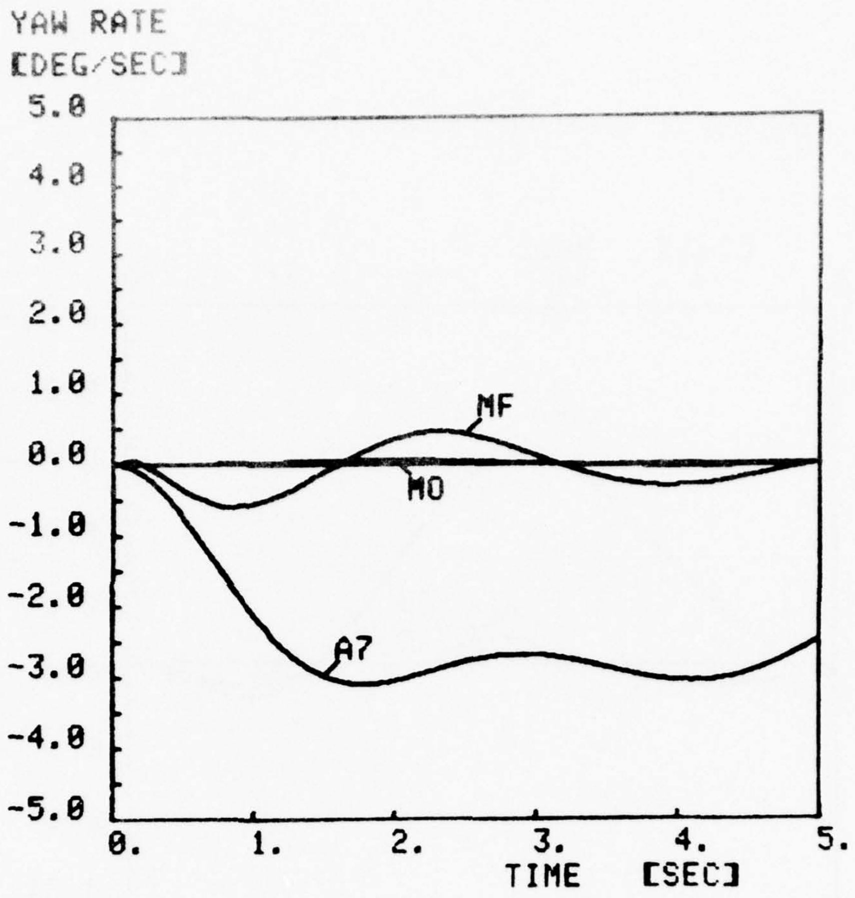


Fig. 16. Case 2 Aircraft Responses in Yaw Rate

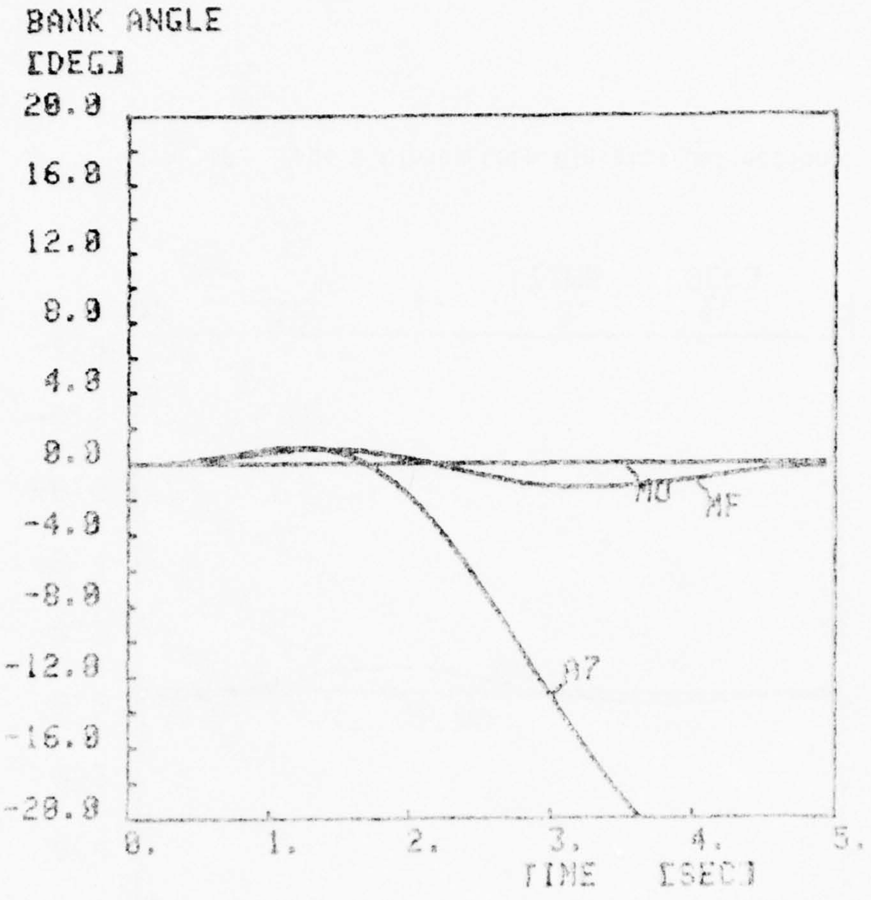


Fig. 17. Case 2 Aircraft Responses in Bank Angle

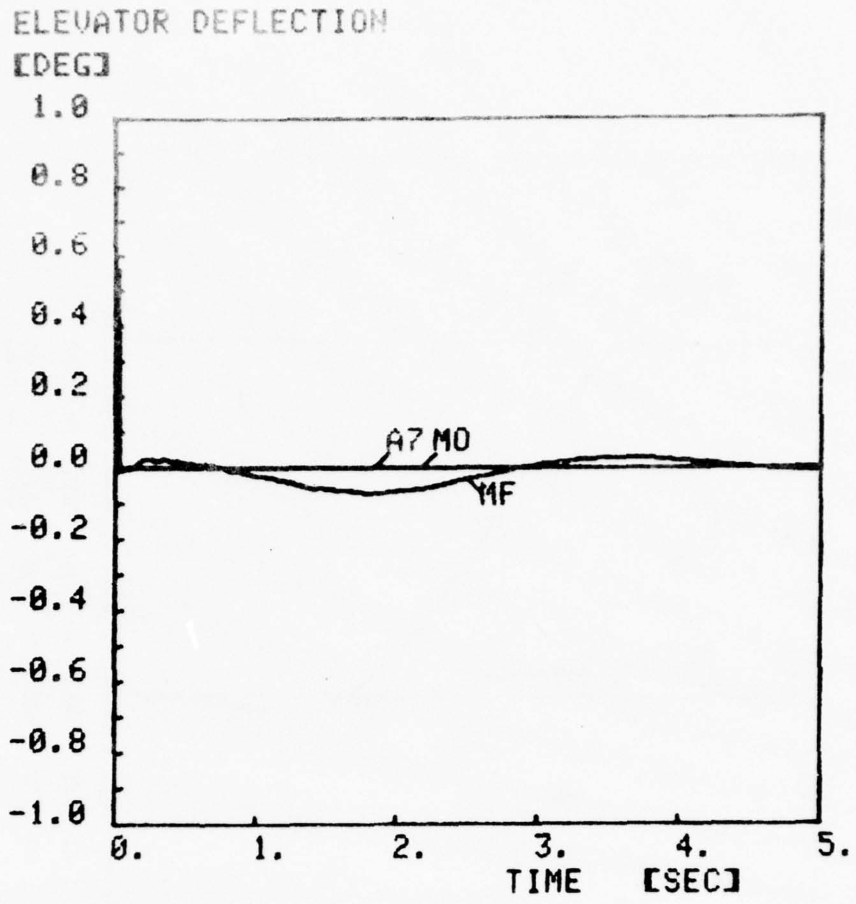


Fig. 18. Case 2 Closed Loop Elevator Deflection

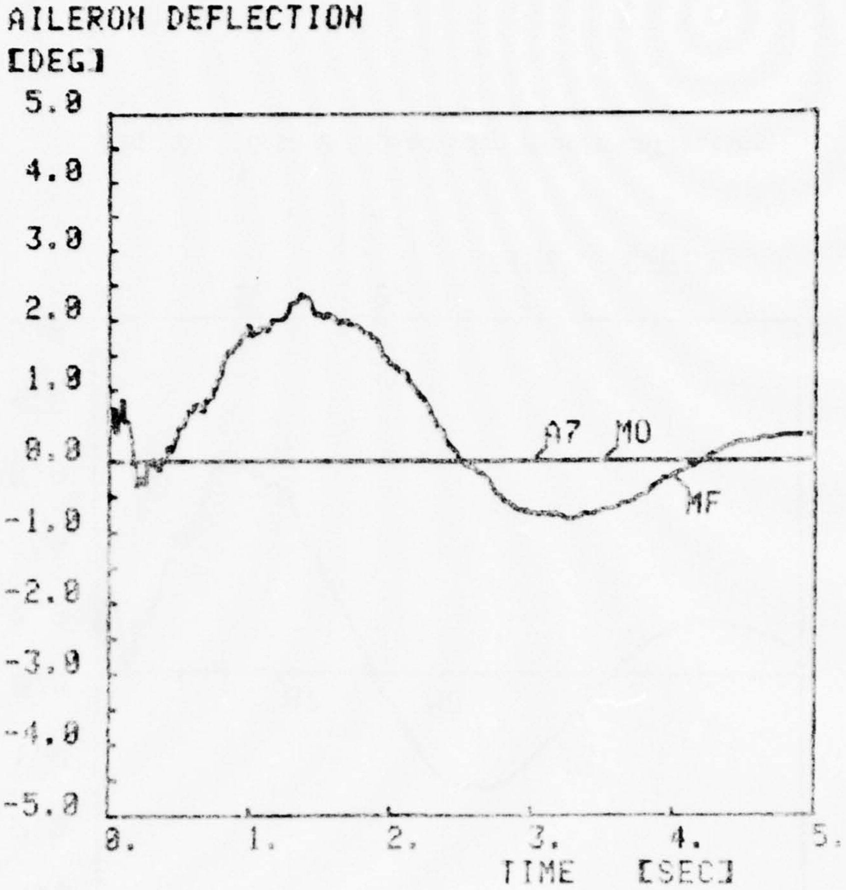


Fig. 19. Case 2 Closed Loop Aileron Deflection

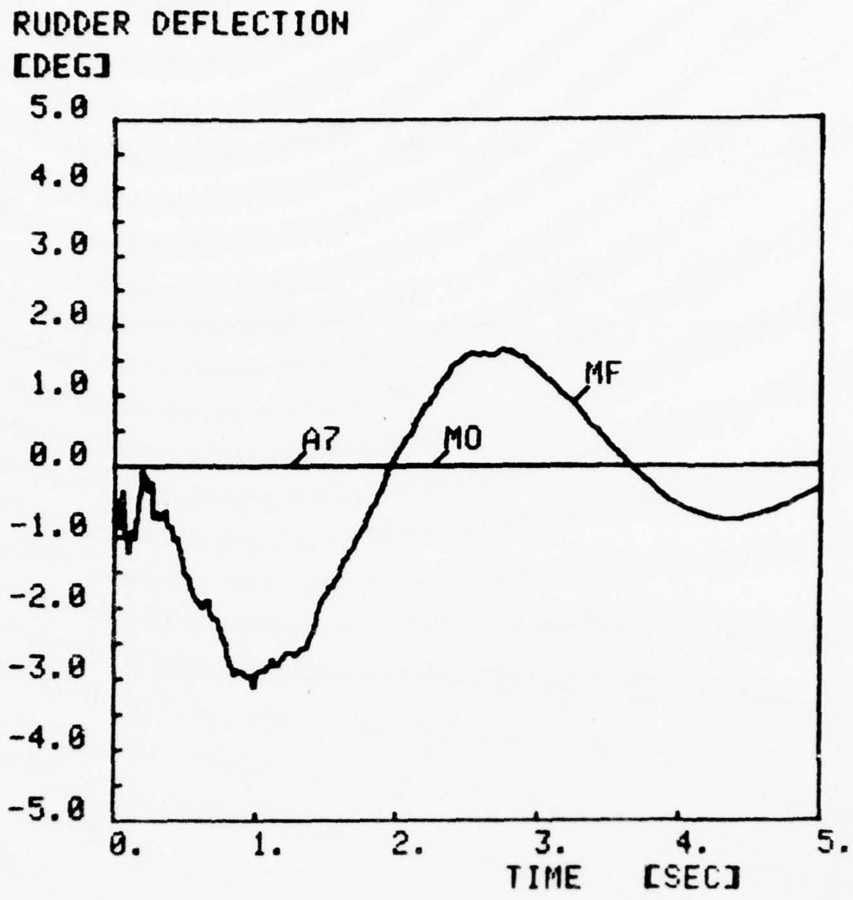


Fig. 20. Case 2 Closed Loop Rudder Deflection

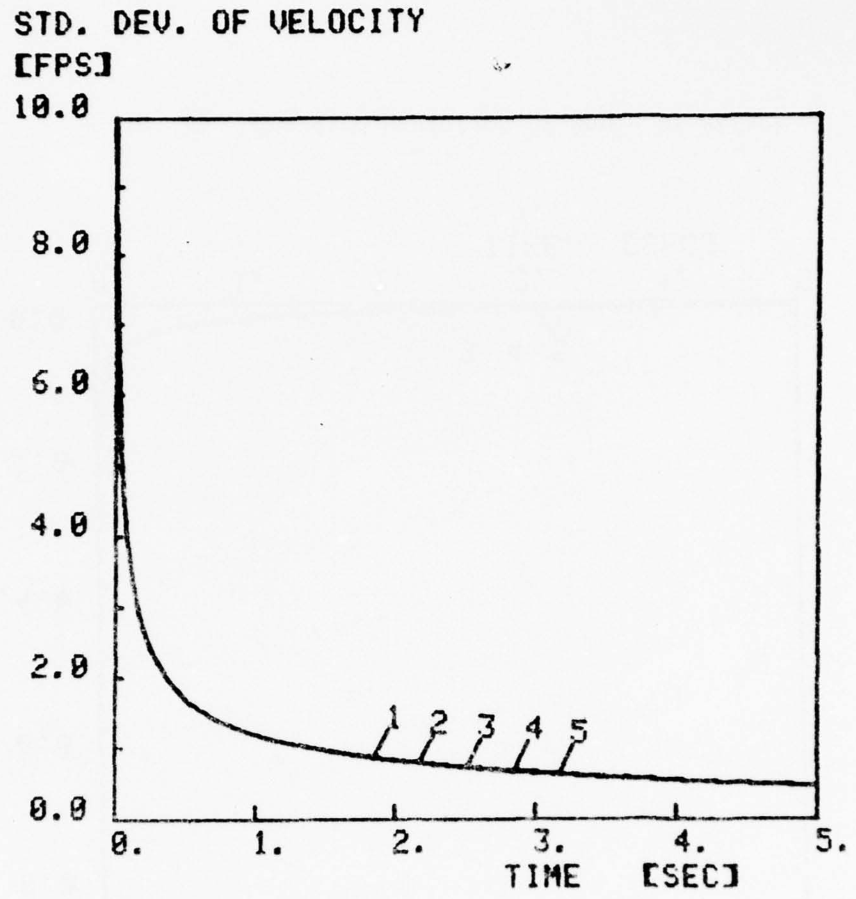


Fig. 21. Standard Deviation of Velocity

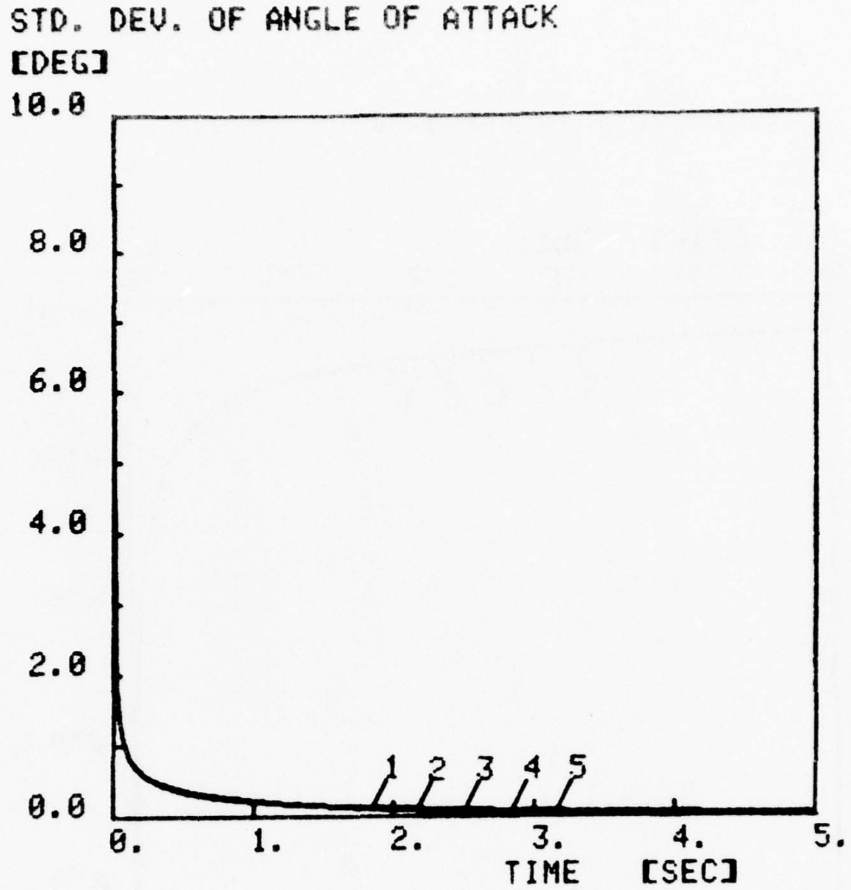


Fig. 22. Standard Deviation of Angle of Attack

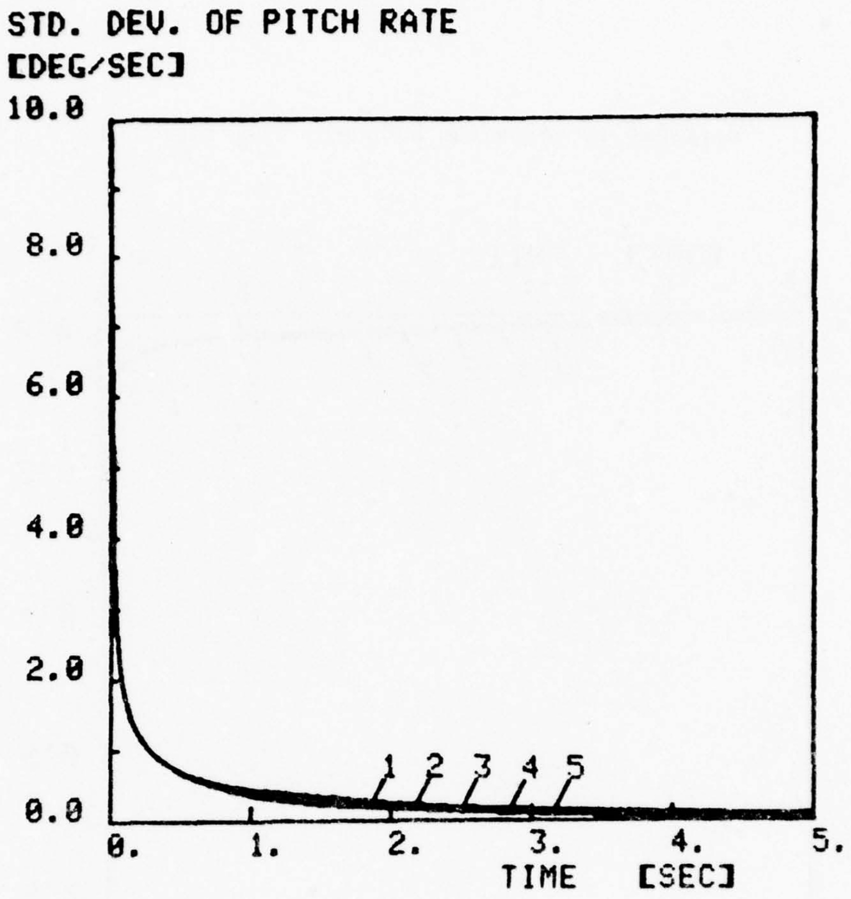


Fig. 23. Standard Deviation of Pitch Rate

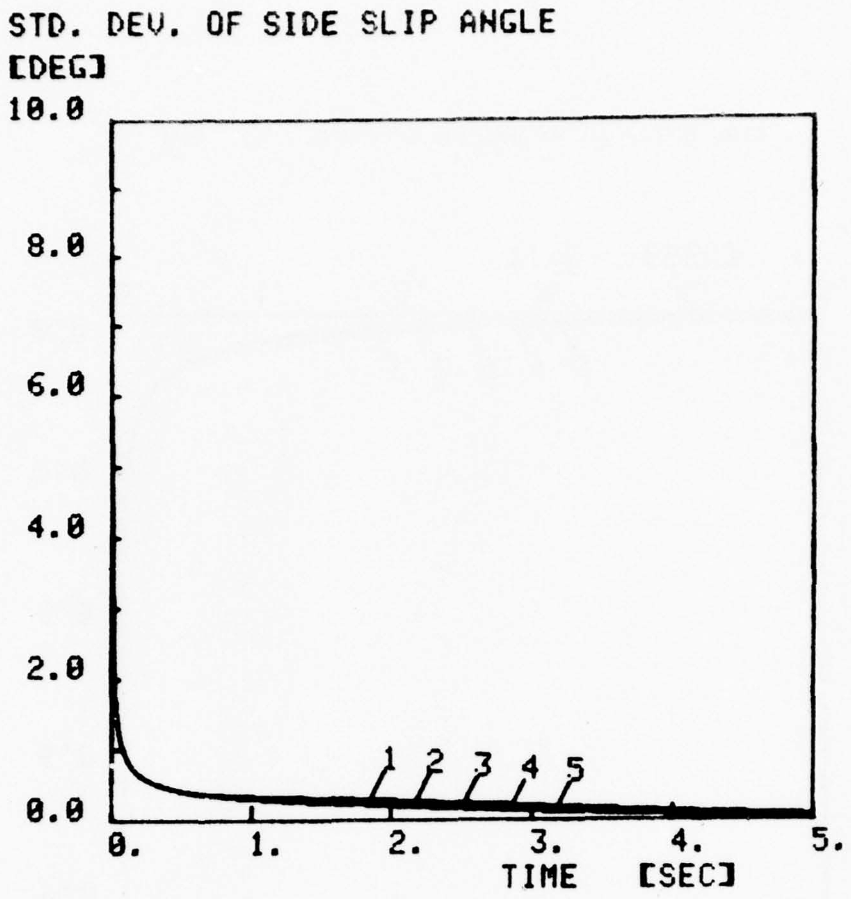


Fig. 24. Standard Deviation of Sideslip

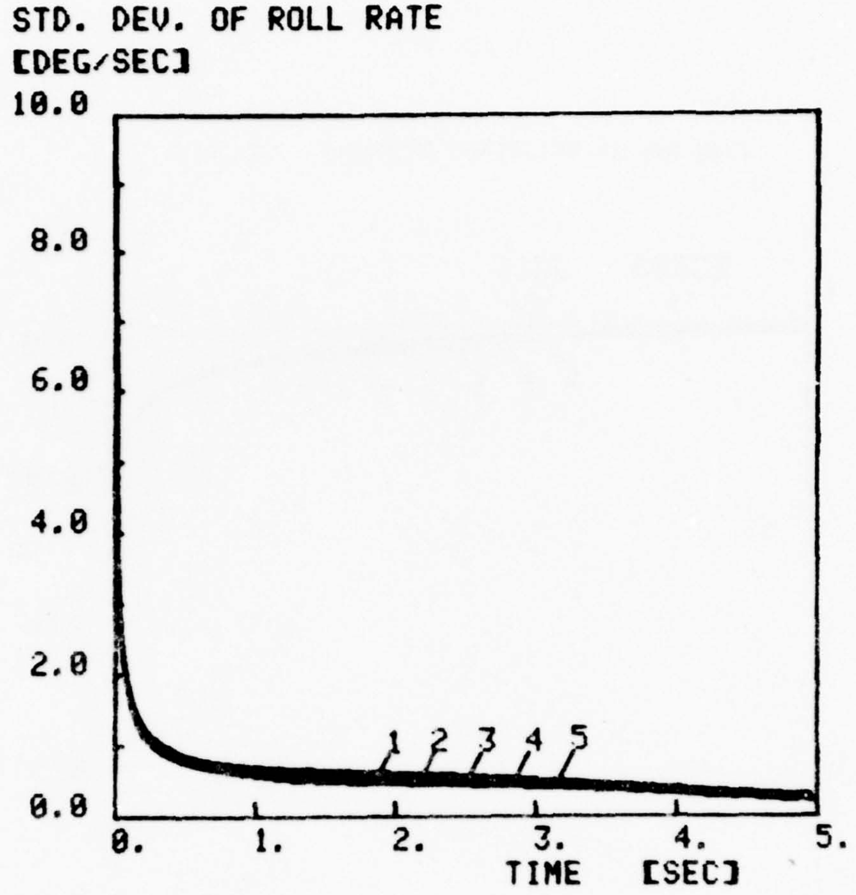


Fig. 25. Standard Deviation of Roll Rate

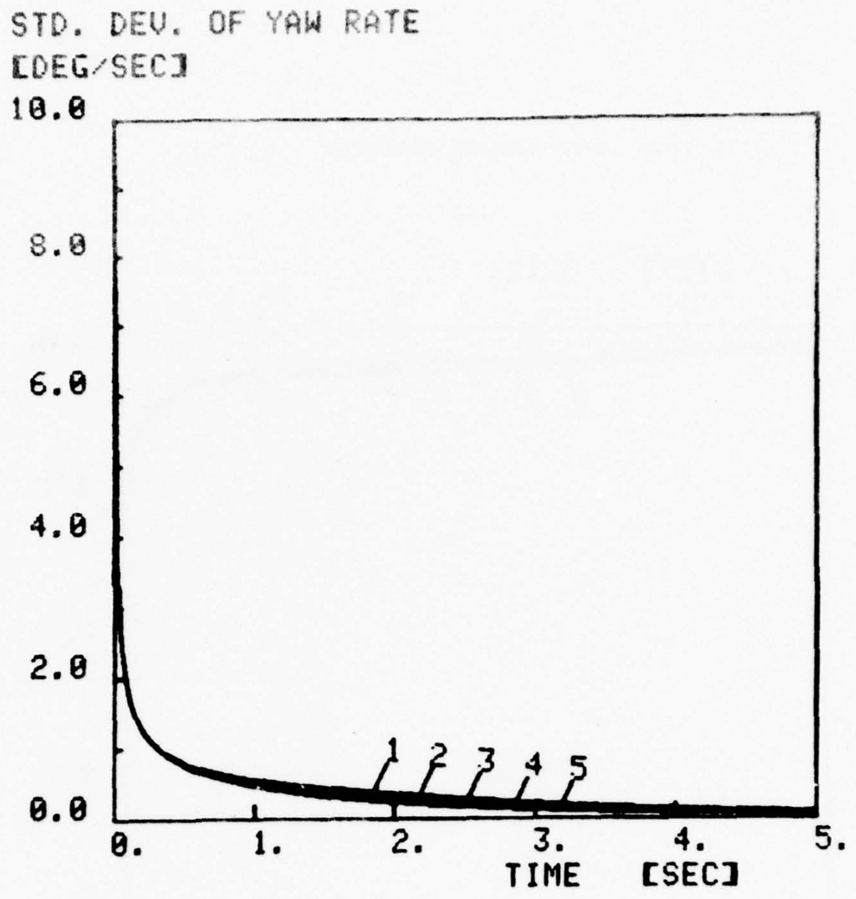


Fig. 26. Standard Deviation of Yaw Rate

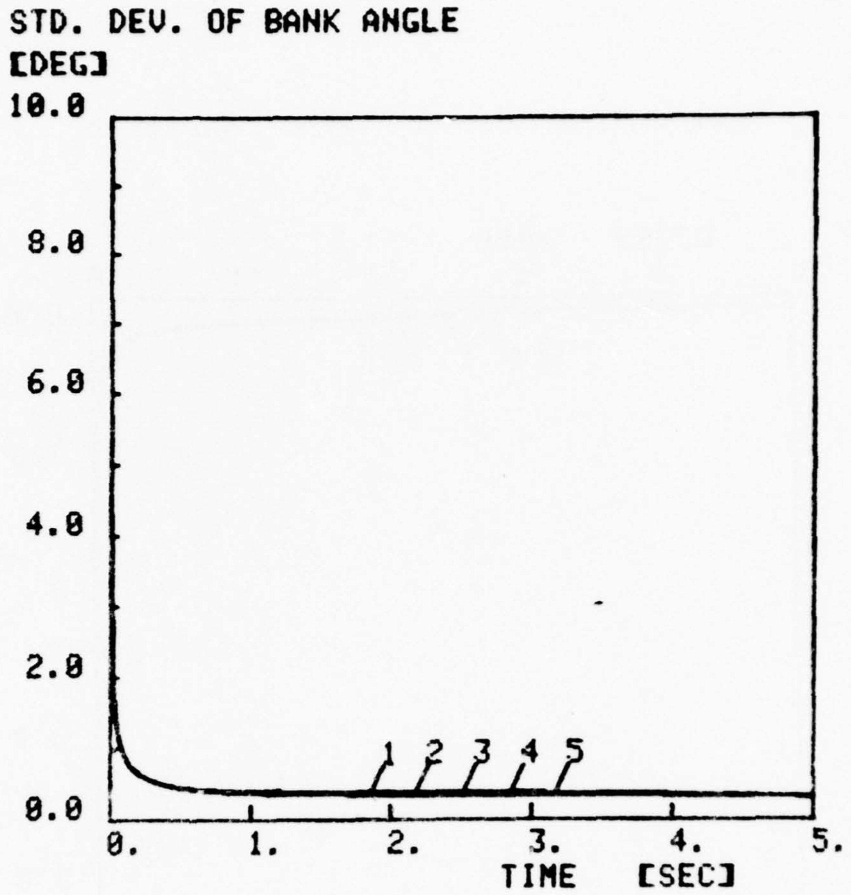


Fig. 27. Standard Deviation of Bank Angle

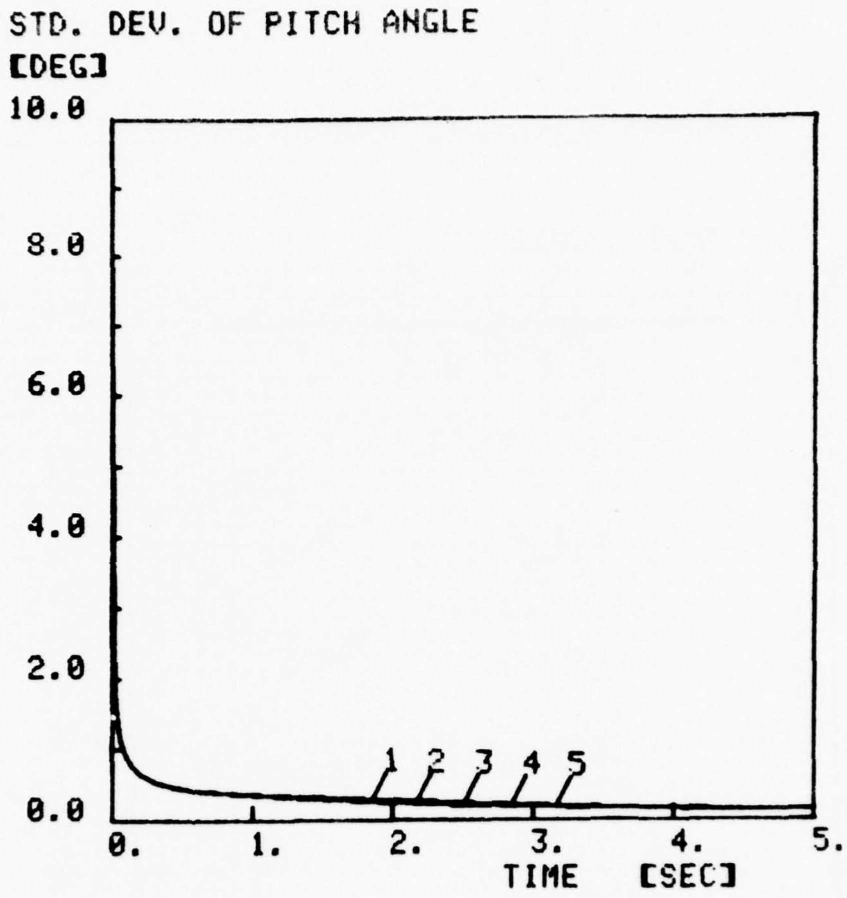


Fig. 28. Standard Deviation of Pitch Angle

APPENDIX A
Equations of Motion

The equations of motion for the aircraft are

$$\dot{x} = Ax + bu \tag{A.1}$$

where

$$A = \begin{bmatrix} X_u & X_w U_o & 0 & x_\beta & 0 & 0 & g\beta_o \cos\theta_o & -g\cos\gamma_o \\ Z_u/U_o & Z_w & +1 & 0 & -\beta_o \cos\alpha_o & -\beta_o \sin\alpha_o & 0 & 0 \\ 0 & \begin{matrix} M_\alpha \\ +M_{\dot{w}} Z_\alpha \end{matrix} & \begin{matrix} M_a \\ +M_{\dot{\alpha}} \end{matrix} & M_\beta & \begin{matrix} -r_o(\ ) \\ -p_o(\ ) \end{matrix} & \begin{matrix} +r_o(\ ) \\ -p_o(\ ) \end{matrix} & 0 & 0 \\ 0 & Y_\alpha/U_o & 0 & Y_v & \sin\alpha_o & -\cos\alpha_o & \frac{g'}{U_o}\cos\theta_o & \frac{g}{U_o}\beta_o \sin\gamma \\ 0 & L'_\alpha & 0 & L'_\beta & L'_p & L'_r & 0 & 0 \\ 0 & N'_\alpha & 0 & N'_\beta & N'_p & N'_r & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & \tan\theta_o & 0 & \frac{r_o}{\cos^2\theta_o} \\ 0 & 0 & +1 & 0 & 0 & 0 & r_o & 0 \end{bmatrix} \tag{A.2}$$

and

$$B = \begin{bmatrix} X_{\delta_e} & 0 & X_{\delta_r} \\ Z_{\delta_e}/U_o & 0 & 0 \\ M_{\delta_e} & 0 & 0 \\ Y_{\delta_e}/U_o & 0 & Y_{\delta_r}/U_o \\ L'_{\delta_e} & L'_{\delta_a} & L'_{\delta_r} \\ N'_{\delta_e} & N'_{\delta_a} & N'_{\delta_r} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (A.3)$$

with the state vector as given in equation (1) and where X_i, Y_i, Z_i is the expression for $\partial(\ )/\partial_i$, u_v is the perturbed total linear velocity, U_0 is the free stream velocity, r_0 is *nominal yaw rate*, p_0 is the nominal roll rate, g is gravity, M_i is $\partial M/\partial i$, γ_0 is the flight path angle, and δe, δa, and δr are the elevator, aileron and rudder deflections, respectively. The aerodynamic coefficients may be found in references [11, 12].

The values for the matrices about the α, β point chosen for the simulations are

$$A = \begin{bmatrix} -.0634 & -22.68 & 0 & -5.766 & 0 & 0 & +3.187 & -32.024 \\ -.00087 & -.323 & +1. & 0 & -.0995 & -.0338 & 0 & 0 \\ 0 & -3.577 & -.386 & -.9 \times 10^{-7} & -.00818 & +.0025 & 0 & 0 \\ 0 & +.0122 & 0 & -.1062 & +.3216 & -.9469 & +.1166 & +.0129 \\ 0 & +3.09 & 0 & -4.45 & -.849 & +.3323 & 0 & 0 \\ 0 & -1.486 & 0 & -.1885 & +.0193 & -.1276 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1. & -.3396 & 0 & -.0116 \\ 0 & 0 & +1. & 0 & 0 & 0 & +.0104 & 0 \end{bmatrix} \qquad (A.4)$$

$$B = \begin{bmatrix} -0.1025 & 0 & 0.698 \\ -0.057 & 0 & 0 \\ -2.92 & 0 & 0 \\ -0.0037 & 0 & 0.0255 \\ -0.292 & 0.431 & 1.4 \\ 0.1095 & 0.031 & -0.998 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (A.5)$$

For the C matrix in the measurement equation (compare Eq. 2, p. 3), a unit matrix (8 x 8) was used in the simulation.

APPENDIX B
Model Following Program

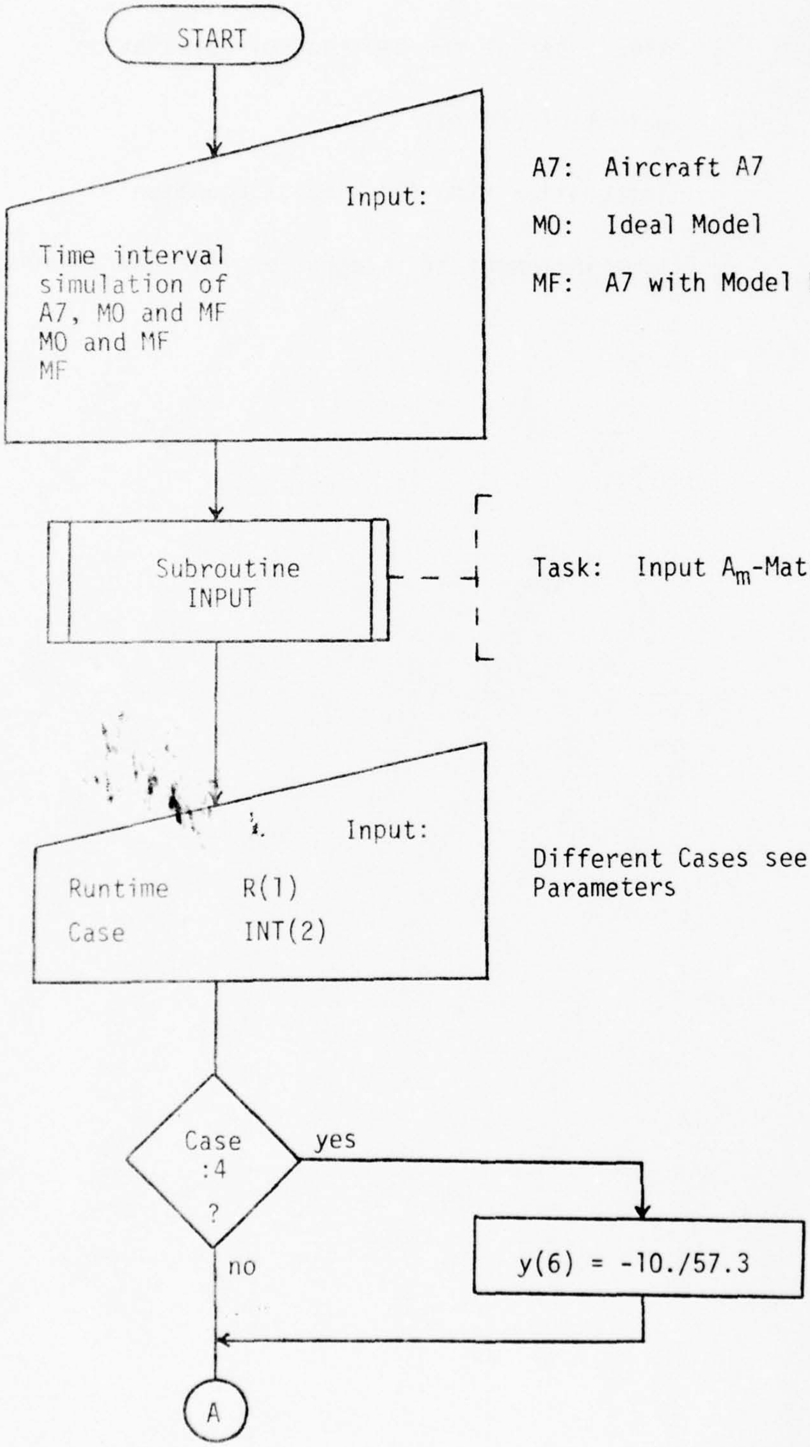
-Part 1-
Control Calculations and Flight Simulations

1. Important Parameters

INT(1):	-		not used
INT(2):	(Case)	1:	open loop, ramp δ_e
		2:	close loop $\Theta \rightarrow \delta_e$, step Θ_c
		3:	as case 2 & $u \rightarrow \delta_T$
		4:	open loop $r(t=0) = -10$ deg/sec
		5:	open loop $q(t=0) = 5$ deg/sec
INT(3):	MODE	-2:	run preparation
		-1:	
		0:	simulation run
		+1:	end of simulation
INT(4):	NO	1:	A7, MO & MF simulation
		2:	MO & MF simulation
		3:	MF simulation
	NF	1:	A7 output file
		2:	MO output file
		3:	MF output file
INT(5):		0:	A7 derivatives const
		1:	nonlinear simulation (in older program version only)
INT(6):	KEN	0:	first call of several subroutines
		1:	not first call of several subroutines
INT(7):	IHALF		max number of bisections of the initial time increment in $\hat{x}$ - calculation
INT(8):	-		not used
INT(9):	-		not used

INT(10):	-	not used
R(1)	XM	runtime
R(2)	DELTIM	time interval for new control calculation
R(3)	W	weight of control
R(4)		integration time for Riccati equation
R(5)		time increment of integration (Riccati equation)

2. Flow Chart Program, Part 1



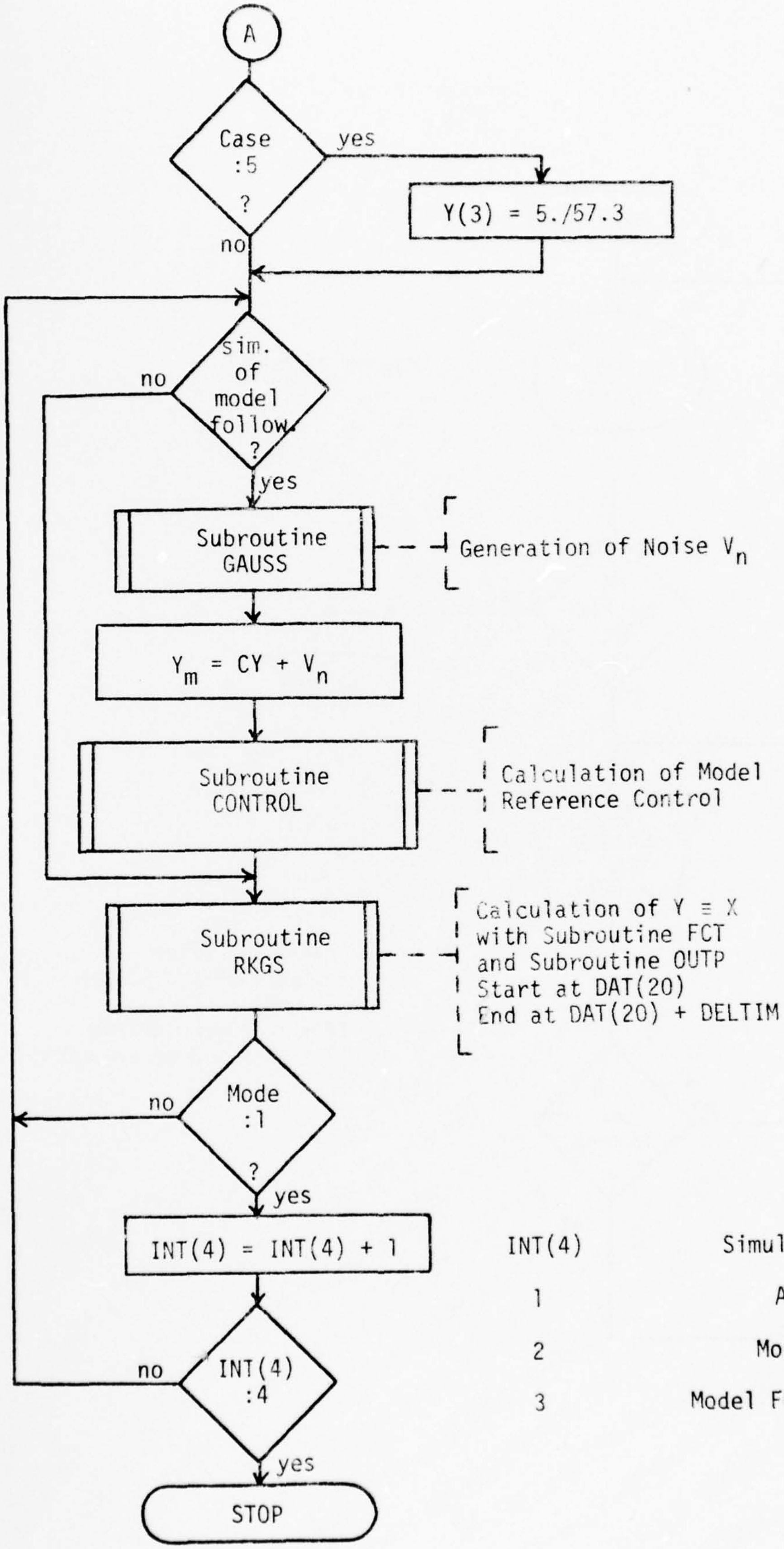
A7: Aircraft A7

MO: Ideal Model

MF: A7 with Model Reference Control

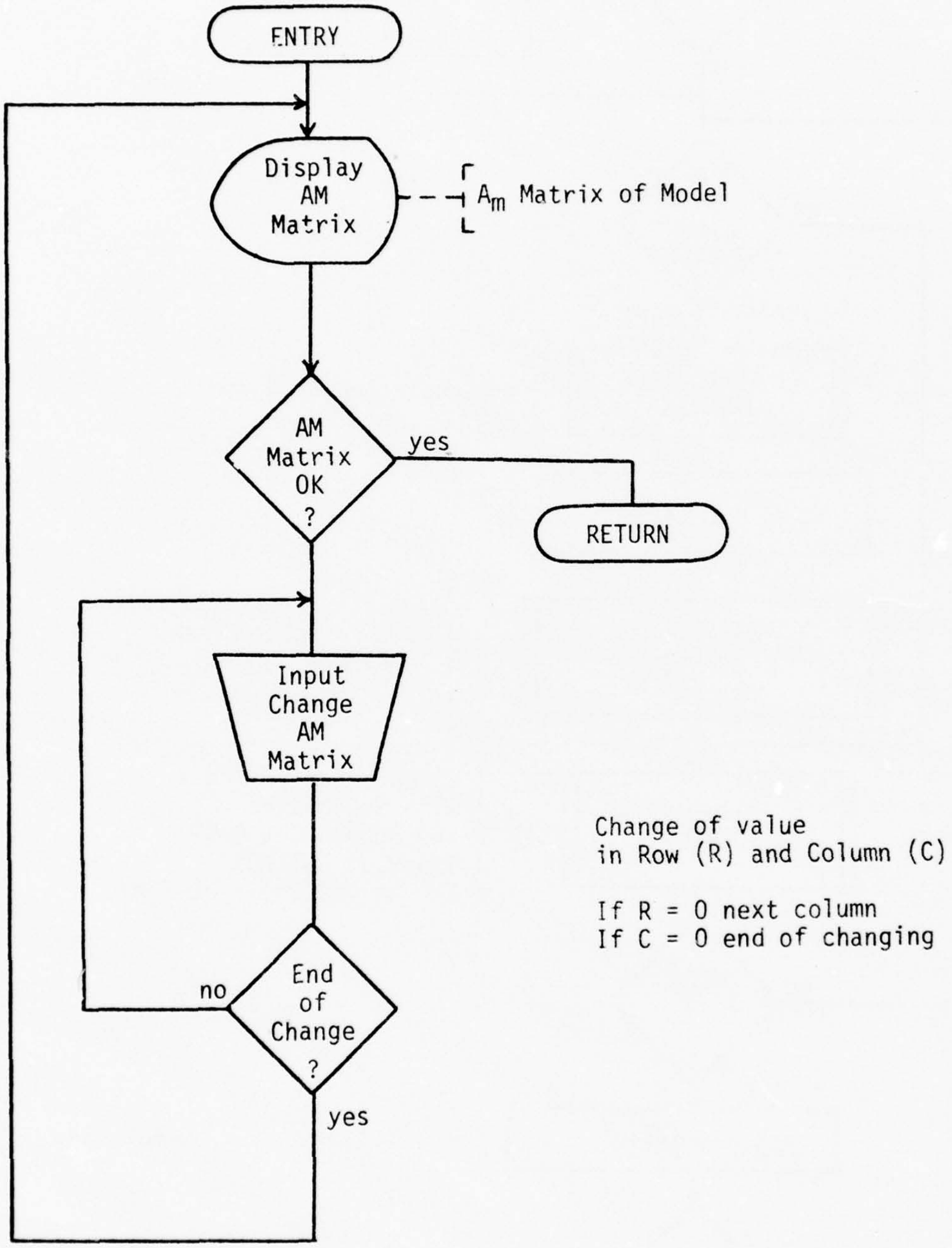
Task: Input A_m-Matrix for Model

Different Cases see Important Parameters


A
Case :5 ?
yes
no
Y(3) = 5./57.3
sim. of model follow. ?
no
yes
Subroutine GAUSS
Generation of Noise V_n
$Y_m = CY + V_n$
Subroutine CONTROL
Calculation of Model Reference Control
Subroutine RKGS
Calculation of $\dot{Y} \equiv \dot{X}$ with Subroutine FCT and Subroutine OUTP Start at DAT(20) End at DAT(20) + DELTIM
Mode :1 ?
no
yes
INT(4) = INT(4) + 1
INT(4) :4
no
yes
STOP

INT(4)	Simulation
1	A7
2	Model
3	Model Following

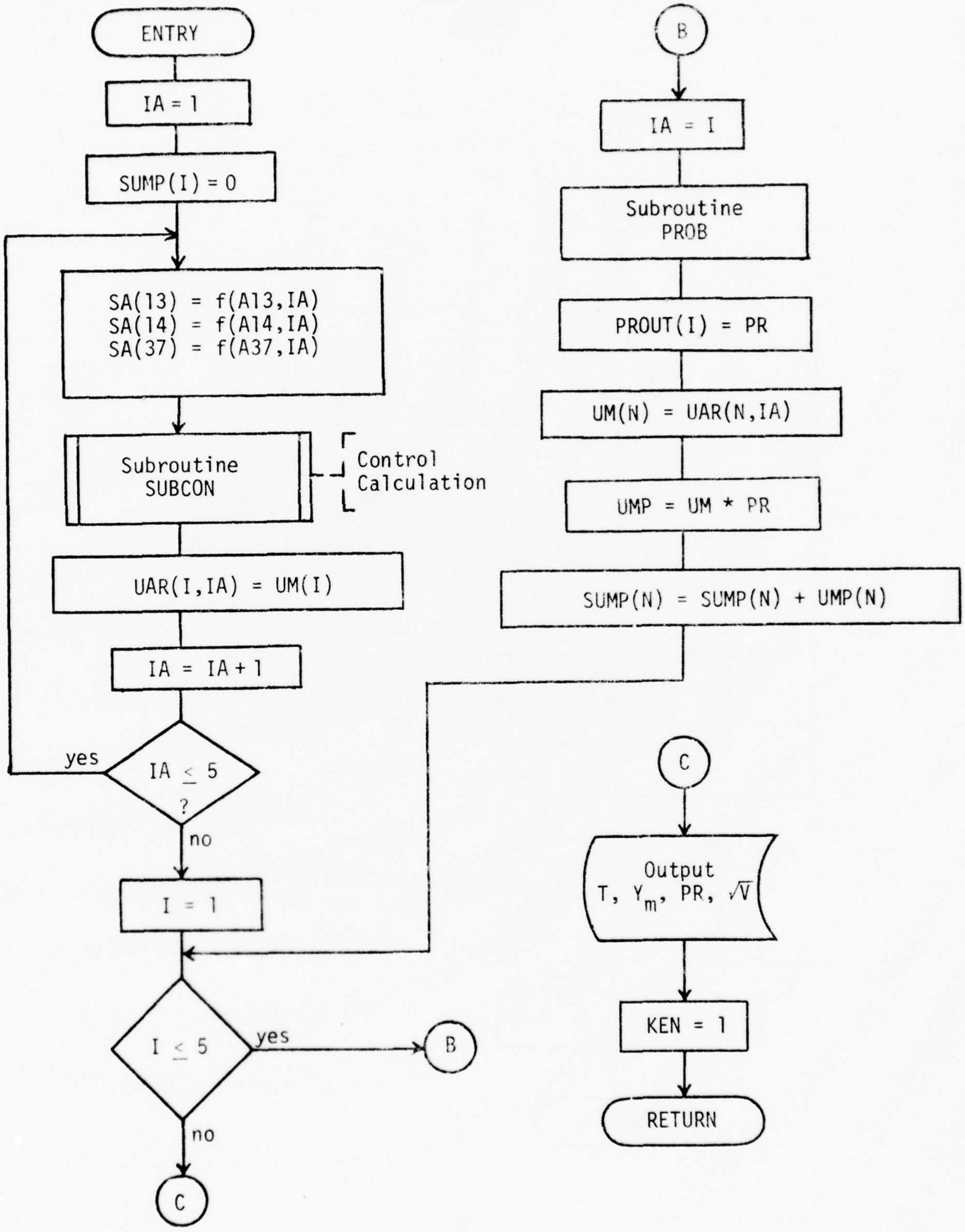
Subroutine INPUT



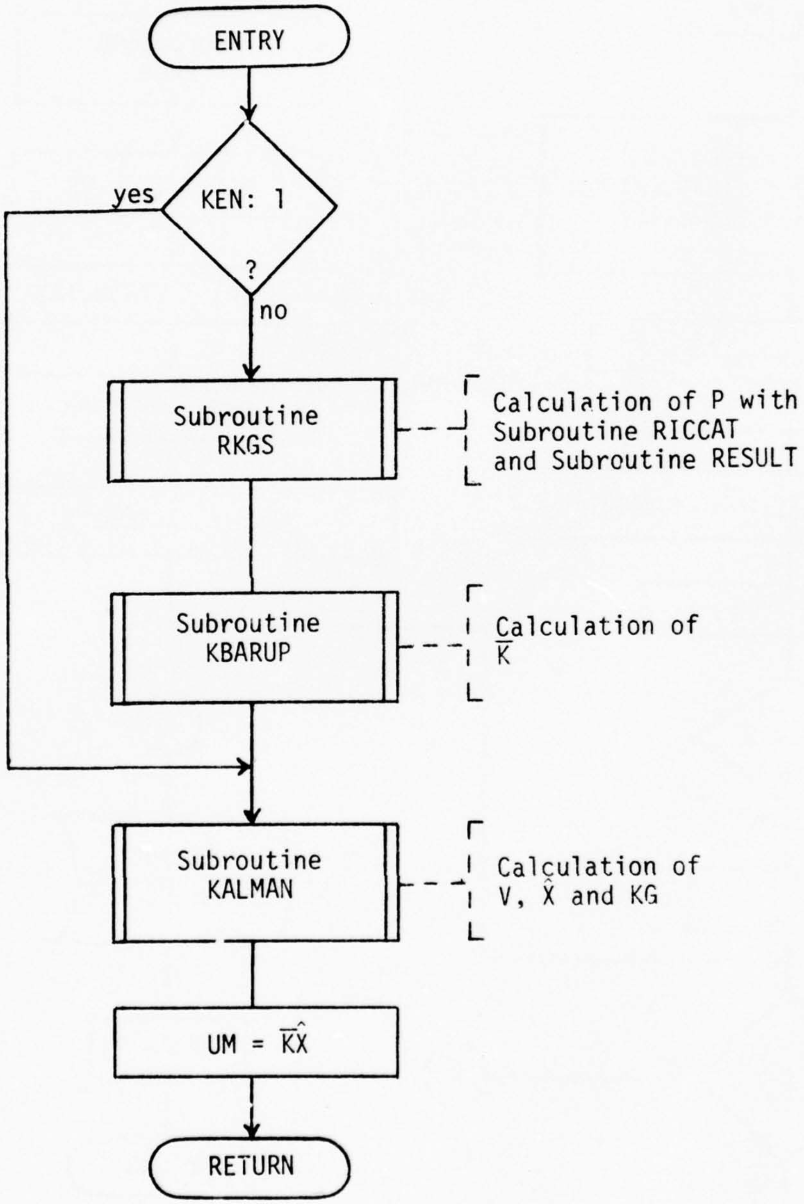
Change of value
in Row (R) and Column (C)

If R = 0 next column
If C = 0 end of changing

Subroutine
CONTRL

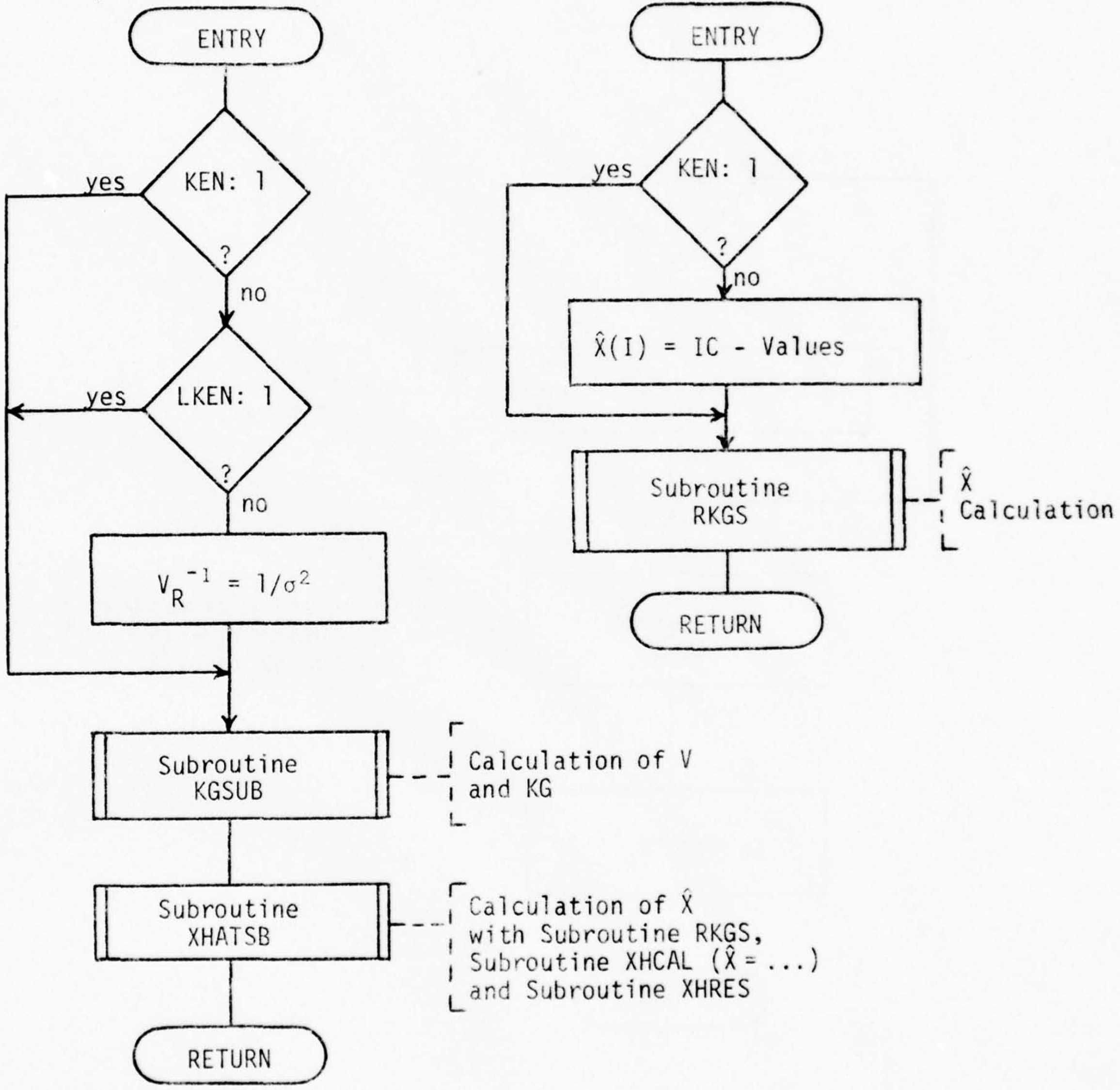


Subroutine
SUBCON

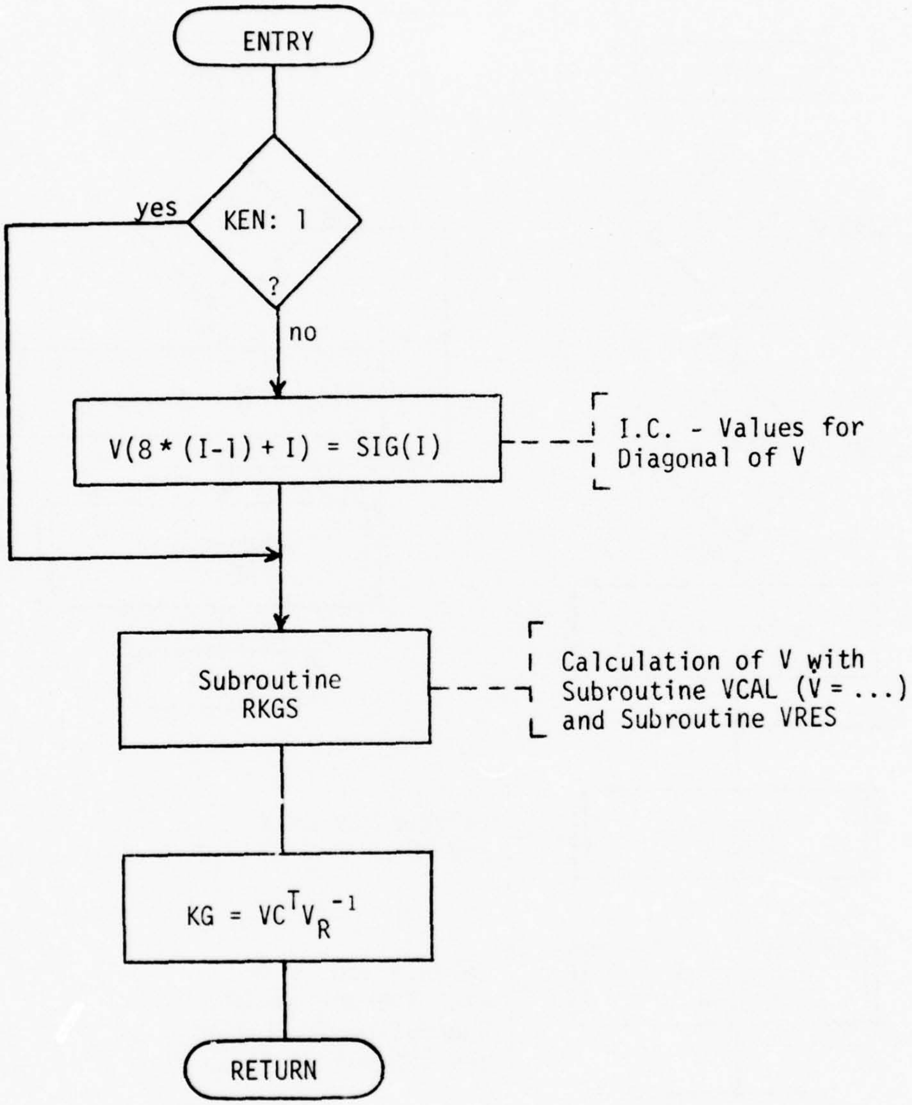

ENTRY
KEN: 1
?
yes
no
Subroutine RKGS
Calculation of P with Subroutine RICCAT and Subroutine RESULT
Subroutine KBARUP
Calculation of $\overline{K}$
Subroutine KALMAN
Calculation of V, $\hat{X}$ and KG
UM = $\overline{K}\hat{X}$
RETURN

Subroutine
KALMAN

Subroutine
XHATSB



Subroutine KGSUB

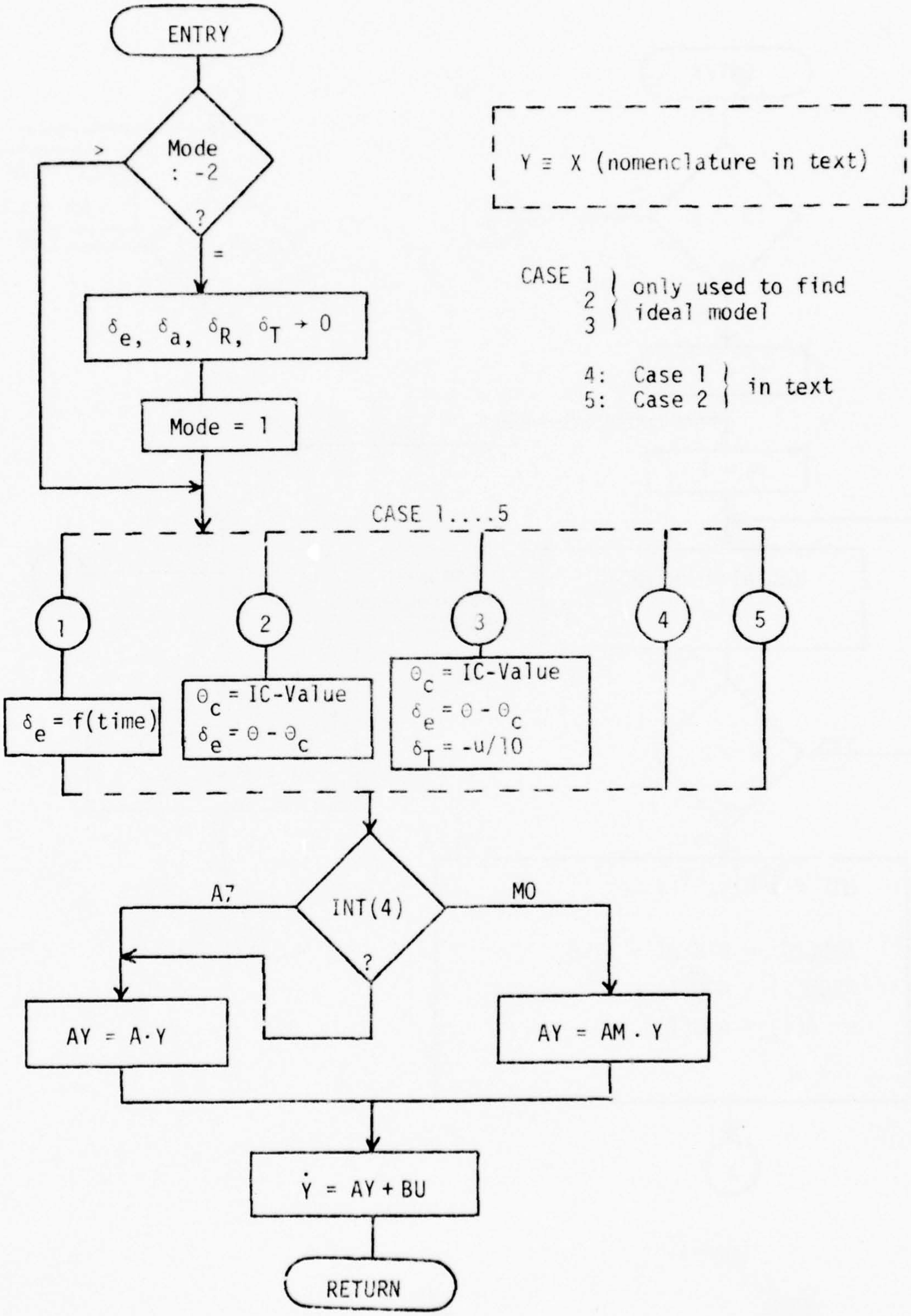


I.C. - Values for Diagonal of V

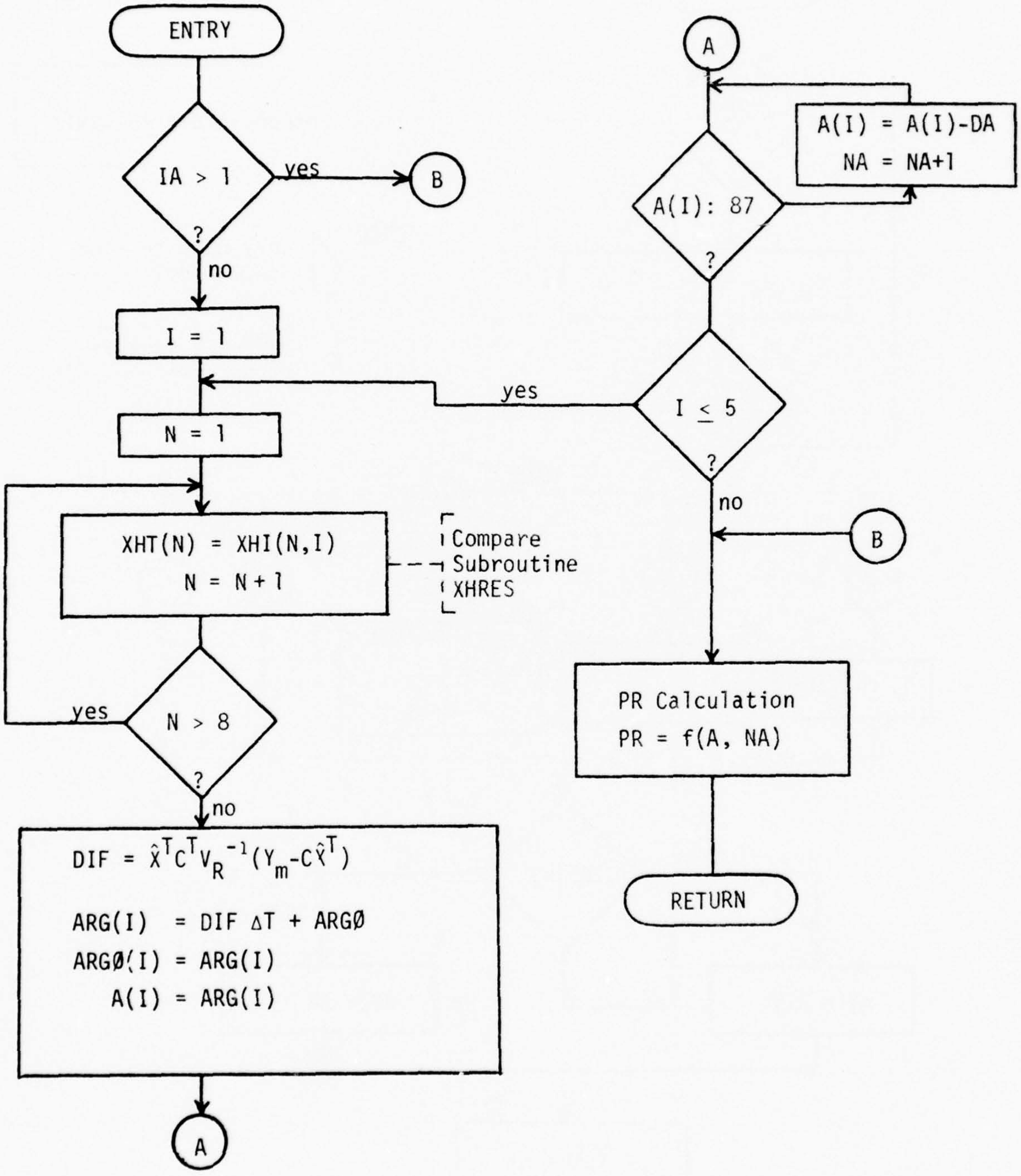
Calculation of V with Subroutine VCAL (V = ...) and Subroutine VRES

Subroutine

FCT



Subroutine
PROB



3. Program Comments and Source Listing

The variable names correspond closely to the notation in the text. The probability density function (compare Equation 16, page 8) was calculated (with $P(\mu_{2_j}) = 1$) in the following way (short notation):

$$P_r(i) = \frac{P(i)e^{A(i)}}{\sum_{j=1}^{5} P(j)e^{a(j)}}$$

with

$$A = \int_{t_o}^{t} \hat{X}^T C^T V_R^{-1} Y_m dt - \int_{t_o}^{t} ||C\hat{X}||^2_{V_R^{-1}} dt$$

$$= \int_{t_o}^{t} \hat{X}^T C^T V_R^{-1} Y_m dt - \int_{t_o}^{t} \hat{X}^T C^T V_R^{-1} C\hat{X}\, dt$$

$$= \int_{t_o}^{t} \hat{X}^T C^T V_R^{-1} (Y_m - C\hat{X})\, dt$$

To prevent overflow for longer simulation time

$$A(i) = A_n(i) + n(i)\, D_a$$

$$D_a = \text{const},\ A_n < 87.0$$

was introduced. Therefore,

$$P_r(i) = \frac{1}{\text{sum}(i)}$$

with

$$\text{sum}(i) = \sum_{j=1}^{5} \frac{P(j)}{P(i)}\, e^{A_i}$$

and

$$A_i = A_n(j) - A_n(i) + \left(n(j) - n(i)\right) D_a$$

The program writes the results on the disk.

File name:

FOR001.DAT for A7

FOR002.DAT for Ideal Model (MO)

FOR003.DAT for Model Following (MF)

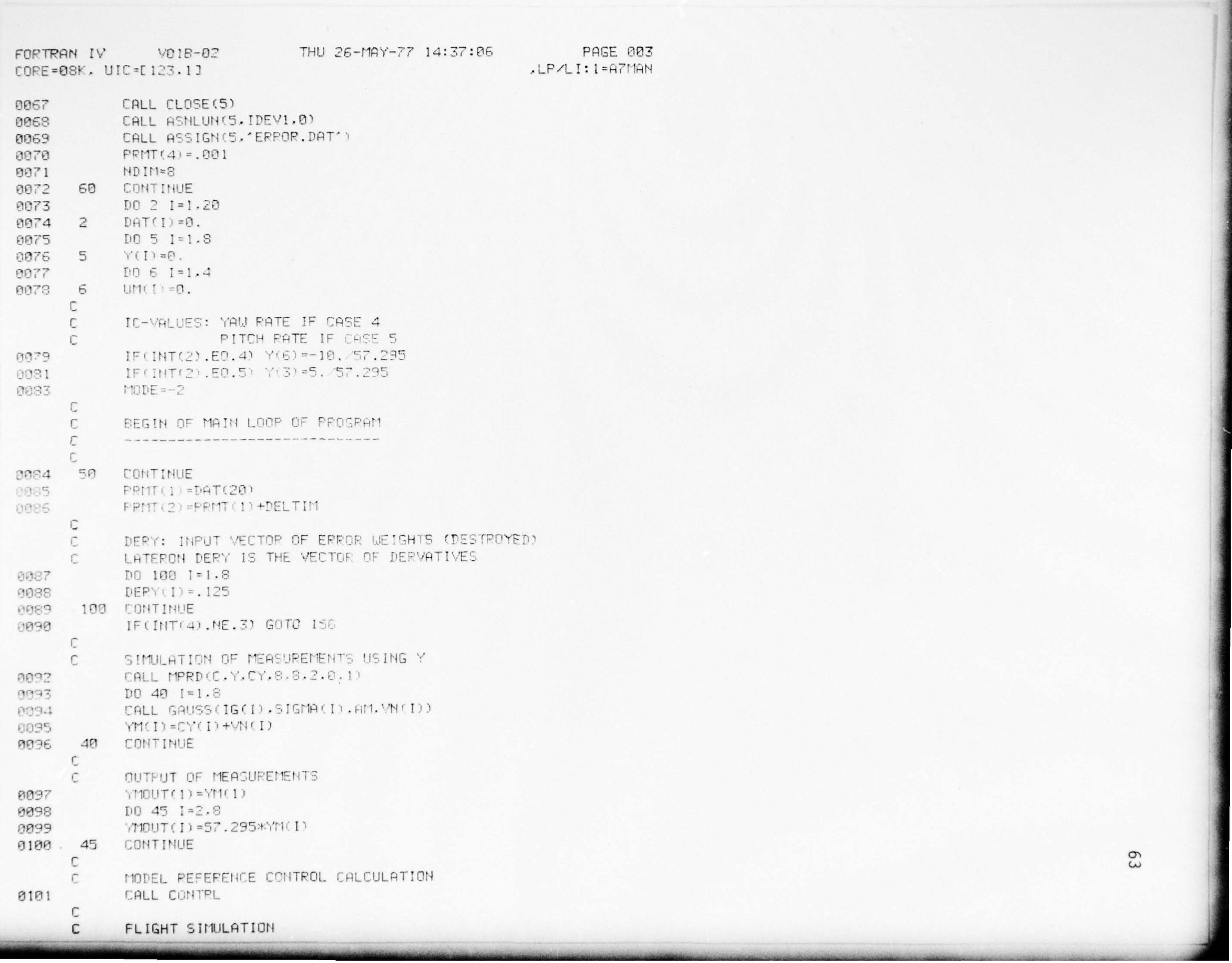
The following pages show a typical dialog.

```
INS A7MAN/PRI=40
>RUN A7MAN
>
 TERMINAL OUTPUT (Y/N):N
T1:.02
DELTAT1:.002
T2:20.
DELTAT2:.5

SIMULATION OF
A7, MO, MF : NO=1
MO, MF     : NO=2
MF         : NO=3
NO=1
 -0.0634-22.6800  0.0000 -5.7660  0.0000  0.0000  3.1870-32.0240
 -0.0009 -0.3230  1.0000  0.0000 -0.0995 -0.0338  0.0000  0.0000
  0.0000 -3.5770 -0.3860 -0.0000 -0.0082  0.0025  0.0000  0.0000
  0.0000  0.0122  0.0000 -0.1862  0.3216 -0.9469  0.1166  0.0129
  0.0000  3.0900  0.0000 -4.4500 -0.8490  0.3323  0.0000  0.0000
  0.0000 -1.4860  0.0000 -0.1885  0.0193 -0.1276  0.0000  0.0000
  0.0000  0.0000  0.0000  0.0000  1.0000  0.3397  0.0000 -0.0116
  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0184  0.0000
AM-MATRIX OK? (Y/N): N
```

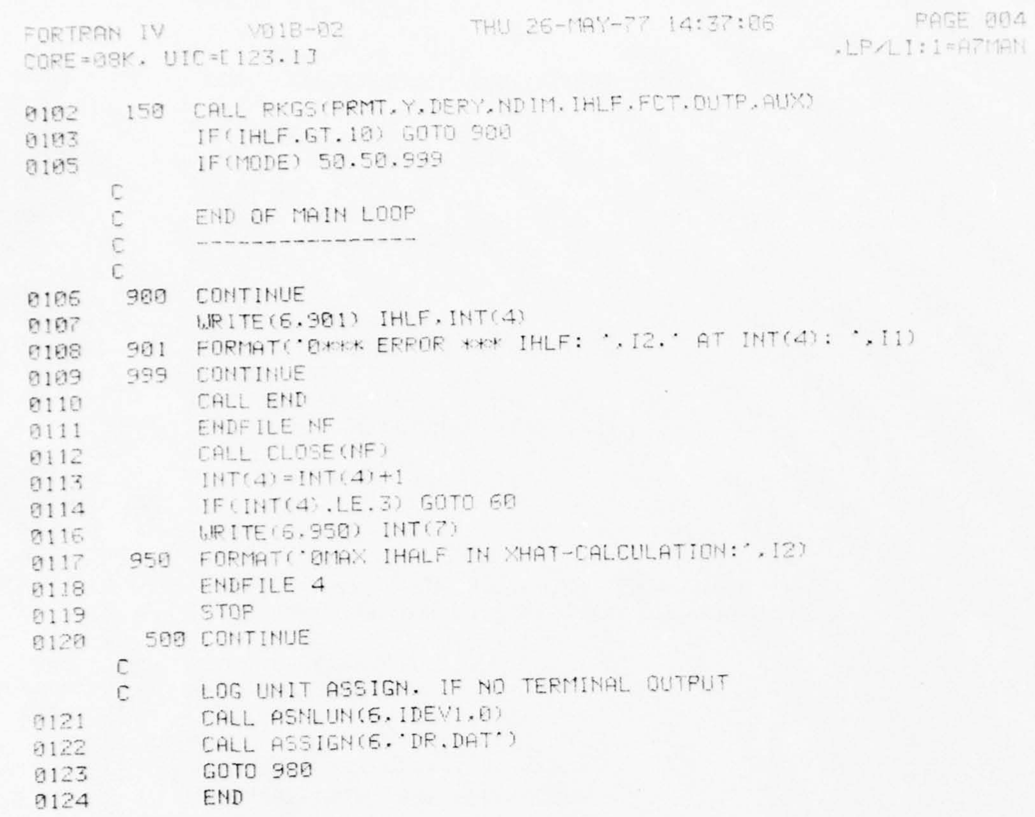
```
CHANGE OF VALUE IN ROW (R) AND COLUMN (C)

IF R=0 NEXT COLUMN,
IF C=0 END OF CHANGING
C: 2
R: 5
AM(5,2)= 0.
R: 6
AM(6,2)= 0.
R:
C: 4
R: 5
AM(5,4)= -1.
R: 6
AM(6,4)= 1.5
R:
C: 7
R: 5
AM(5,7)= -.5
R:
C:
 -0.0634-22.6800  0.0000 -5.7660  0.0000  0.0000  3.1870-32.0240
 -0.0009 -0.3230  1.0000  0.0000 -0.0995 -0.0338  0.0000  0.0000
  0.0000 -3.5770 -0.3860 -0.0000 -0.0082  0.0025  0.0000  0.0000
  0.0000  0.0122  0.0000 -0.1062  0.3216 -0.9469  0.1166  0.0129
  0.0000  0.0000  0.0000 -1.0000 -0.8490  0.3323 -0.5000  0.0000
  0.0000  0.0000  0.0000  1.5000  0.0193 -0.1276  0.0000  0.0000
  0.0000  0.0000  0.0000  0.0000  1.0000  0.3397  0.0000 -0.0116
  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0104  0.0000
AM-MATRIX OK? (Y/N): Y
RUN TIME: 5.
CASE: 4
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:06         PAGE 001
CORE=08K. UIC=[123.1]                                         .LP/LI:1=A7MAN

          C MODEL FOLLOWING PROGRAM
          C -----------------------
          C       PART 1
          C       ------
          C       INPUT AND CALCULATION
          C       VERSION 5/26
          C
          C       IN THIS PART OF THE PROGRAM MANY SUBROUTINES
          C       OF THE SCIENTIFIC SUBROUTINE PACKAGE (SEE
          C       RSX-11M MANUAL) WERE USED.
          C
0001              COMMON /A7/DAT(20).INT(10).R(5)
0002              COMMON /MATR/SA(64).SB(32).SAM(64).UM(4)
0003              COMMON /KAL/XHATK(8).RKG(64).C(8).CT(8).VRIN(8)
0004              COMMON /NOISE/SIGMA(8).YM(8)
0005              COMMON /OUT/YMOUT(8).SVDOUT(8.5)
0006              DIMENSION PRMT(5).Y(8).DERY(8).AUX(8.8)
0007              DIMENSION CY(8).IG(8).VN(8)
0008              EXTERNAL FCT.OUTP
0009              EQUIVALENCE(MODE.INT(3)).(R(2).DELTIM)
0010              EQUIVALENCE(NF.INT(4)).(W.R(3))
0011              DATA IDEV2/'TT'/.JES/'Y'/.IDEV1/'DP'/
0012              DATA C/8*1./
          C
          C       WEIGHTING OF CONTROL
          C       W=1. MODEL REFERENCE CONTROL ONLY. NO PILOT INPUT
          C       W=0. PILOT INPUT ONLY
0013              W=1.
          C
          C       STANDARD DEVIATION OF THE NORMAL DISTRIBUTION
          C       OF NOISE
0014              SIGMA(1)=2.5
0015              SIGMA(2)=.01
0016              SIGMA(3)=.02
0017              SIGMA(4)=.01
0018              SIGMA(5)=.02
0019              SIGMA(6)=.02
0020              SIGMA(7)=.01
0021              SIGMA(8)=.01
0022              DO 111 I=1.8
0023              SIGMA(I)=SIGMA(I)/2.
0024        111   CONTINUE
          C
          C       DIFFERENT INPUTS FOR SPECIAL RUN
0025              WRITE(5.10)
0026         10   FORMAT('$TERMINAL OUTPUT (Y/N):')
0027              READ(5.11) IOUT
0028         11   FORMAT(A1)
0029              IF(IOUT.NE.JES) GOTO 500
0031              WRITE(5.250)
```

```
  250 FORMAT('$NUMBER OF TERMINAL:')
      READ(5.201) NT
      CALL ASNLUN(6.IDEV2.NT)
  980 CONTINUE
      WRITE(5.13)
   13 FORMAT('$T1:')
C
C     DELTIM IS THE TIME BETWEEN TWO CONTROL CALCULATIONS
      READ(5.14) DELTIM
   14 FORMAT(F10.6)
      WRITE(5.15)
   15 FORMAT('$DELTAT1:')
C
C     PRMT(3) IS THE INITIAL TIME INCREMENT IN THE
C     RUNGE-KUTTA SUBROUTINE TO SOLVE THE FLIGHT
C     DYNAMICS EQUATIONS
      READ(5.14) PRMT(3)
      WRITE(5.16)
   16 FORMAT('$T2:')
C
C     R(4) IS THE INTEGRATION TIME TO SOLVE
C     THE RICCATI EQUATIONS
      READ(5.14) R(4)
      WRITE(5.17)
   17 FORMAT('$DELTAT2:')
C
C     R(5) IS THE INITIAL TIME INCREMENT IN THE RUNGE-
C     KUTTA SUBROUTINE TO SOLVE THE RICCATI EQUATIONS
      READ(5.14) R(5)
      WRITE(5.12)
   12 FORMAT('0SIMULATION OF'/
     1    ' A7. MO. MF : NO=1'/
     2    ' MO. MF     : NO=2'/
     3    ' MF         : NO=3'/
     4    '$NO=')
      READ(5.201) INT(4)
      NUMF=INT(4)
      DO 600 I=NUMF.4
      CALL FDBSET(I.'UNKNOWN')
  600 CONTINUE
  201 FORMAT(I1)
 401  IF(INT(6).EQ.0) CALL INPUT
  970 CONTINUE
      WRITE(5.210)
 210  FORMAT('$RUN TIME: ')
      READ(5.211) R(1)
 211  FORMAT(F4.0)
      WRITE(5.220)
 220  FORMAT('$CASE: ')
      READ(5.201) INT(2)
```



```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:37:06          PAGE 003
CORE=08K. UIC=[123.1]                                         ,LP/LI:1=A7MAN

0067           CALL CLOSE(5)
0068           CALL ASNLUN(5.IDEV1.0)
0069           CALL ASSIGN(5.'ERROR.DAT')
0070           PRMT(4)=.001
0071           NDIM=8
0072     60    CONTINUE
0073           DO 2 I=1.20
0074      2    DAT(I)=0.
0075           DO 5 I=1.8
0076      5    Y(I)=0.
0077           DO 6 I=1.4
0078      6    UM(I)=0.
         C
         C     IC-VALUES: YAW RATE IF CASE 4
         C                PITCH RATE IF CASE 5
0079           IF(INT(2).EQ.4) Y(6)=-10./57.295
0081           IF(INT(2).EQ.5) Y(3)=5./57.295
0083           MODE=-2
         C
         C     BEGIN OF MAIN LOOP OF PROGRAM
         C     -----------------------------
         C
0084     50    CONTINUE
0085           PRMT(1)=DAT(20)
0086           PRMT(2)=PRMT(1)+DELTIM
         C
         C     DERY: INPUT VECTOR OF ERROR WEIGHTS (DESTROYED)
         C     LATERON DERY IS THE VECTOR OF DERVATIVES
0087           DO 100 I=1.8
0088           DERY(I)=.125
0089    100    CONTINUE
0090           IF(INT(4).NE.3) GOTO 156
         C
         C     SIMULATION OF MEASUREMENTS USING Y
0092           CALL MPRD(C.Y.CY.8.8.2.0.1)
0093           DO 40 I=1.8
0094           CALL GAUSS(IG(I).SIGMA(I).AM.VN(I))
0095           YM(I)=CY(I)+VN(I)
0096     40    CONTINUE
         C
         C     OUTPUT OF MEASUREMENTS
0097           YMOUT(1)=YM(1)
0098           DO 45 I=2.8
0099           YMOUT(I)=57.295*YM(I)
0100     45    CONTINUE
         C
         C     MODEL REFERENCE CONTROL CALCULATION
0101           CALL CONTRL
         C
         C     FLIGHT SIMULATION
```



```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:06          PAGE 004
CORE=08K. UIC=[123.1]                                          .LP/LI:1=A7MAN

0102     150  CALL RKGS(PRMT.Y.DERY.NDIM.IHLF.FCT.OUTP.AUX)
0103          IF(IHLF.GT.10) GOTO 900
0105          IF(MODE) 50.50.999
        C
        C     END OF MAIN LOOP
        C     ----------------
        C
0106     900  CONTINUE
0107          WRITE(6.901) IHLF.INT(4)
0108     901  FORMAT('0*** ERROR *** IHLF: '.I2.' AT INT(4): '.I1)
0109     999  CONTINUE
0110          CALL END
0111          ENDFILE NF
0112          CALL CLOSE(NF)
0113          INT(4)=INT(4)+1
0114          IF(INT(4).LE.3) GOTO 60
0116          WRITE(6.950) INT(7)
0117     950  FORMAT('0MAX IHALF IN XHAT-CALCULATION:'.I2)
0118          ENDFILE 4
0119          STOP
0120      500 CONTINUE
        C
        C     LOG UNIT ASSIGN. IF NO TERMINAL OUTPUT
0121          CALL ASNLUN(6.IDEV1.0)
0122          CALL ASSIGN(6.'DR.DAT')
0123          GOTO 980
0124          END
```

```
FORTRAN IV      VØ1B-Ø2          THU 26-MAY-77 14:37:1Ø          PAGE ØØ1
CORE=Ø8K, UIC=[123,1]                                     ,LP/LI:1=A7MAN

ØØØ1            SUBROUTINE INPUT
          C     ----------------
          C
ØØØ2            COMMON /A7/DAT(2Ø).INT(1Ø),R(5)
ØØØ3            COMMON /MATR/SA(64).SB(32).SAM(64).UM(4)
ØØØ4            COMMON /K/RBAR(16).S(32).SBT(32).RKBAR(32).P(64)
ØØØ5            COMMON /CON/IA.A13.A14.A37
ØØØ6            DIMENSION DA(8.8).DB(8.4).DAM(8.8)
ØØØ7            DATA JES/'Y'/.NO/'N'/
          C
          C     DA IS THE A-MATRIX OF A7 IN DOUBLE DIMENSIONED STORAGE
ØØØ8            DATA DA/-.Ø634.-.ØØØ87.6*Ø..
               1 -22.68.-.323.-3.577..Ø122.3.Ø9.-1.486,2*Ø..
               2 Ø..1..-.386.4*Ø..1..
               3 -5.766,Ø..-.ØØØØØØØ9.-.1Ø62.-4.45.-.1885,2*Ø..
               4 Ø..-.Ø995.-.ØØ818..3216.-.849..Ø193.1..Ø..
               5 Ø..-.Ø338..ØØ25.-.9469..3323.-.1276..3397.Ø..
               6 3.187.2*Ø...1166.3*Ø...Ø1Ø4.
               7 -32.Ø24.2*Ø...Ø129.2*Ø..-.Ø116.Ø./
ØØØ9            DATA DB/-.1Ø25.-.Ø57.-2.92.-.ØØ37,-.292..1Ø95.2*Ø..
               1 4*Ø...431..Ø31.2*Ø..
               2 .698.2*Ø...Ø255.1.4.-.998.2*Ø..
               3 1Ø..7*Ø./
ØØ1Ø            MOD=2
          C
          C     SA IS THE A-MATRIX ØF A7 IN SINGLE DIMENSIONED STORAGE
ØØ11            CALL ARRAY(MOD.8.8.8.8.SA.DA)
ØØ12            A13=SA(13)
ØØ13            A14=SA(14)
ØØ14            A37=SA(37)
          C
          C     DAM IS THE A-MATRIX OF THE IDEAL MODEL
          C     IN DOUBLE DIMENSIONED STORAGE
ØØ15            CALL MCPY(DA.DAM.8.8.Ø)
ØØ16       1ØØ  CONTINUE
ØØ17            N=8
ØØ18            M=8
ØØ19            WRITE(5.1Ø1) ((DAM(IR.IC).IC=1.M).IR=1.N)
ØØ2Ø       1Ø1  FORMAT(' '.8F8.4)
ØØ21            WRITE(5.11Ø)
ØØ22       11Ø  FORMAT('$AM-MATRIX OK? (Y/N): ')
ØØ23            READ(5.111) KN
ØØ24       111  FORMAT(A1)
ØØ25            IF(KN.NE.JES.AND.KN.NE.NO) GOTO 1ØØ
ØØ27            IF(KN.EQ.JES) GOTO 28Ø
ØØ29            WRITE(5.2Ø1)
ØØ3Ø       2Ø1  FORMAT('ØCHANGE OF VALUE IN ROW (R) AND COLUMN (C)'/
               1    'ØIF R=Ø NEXT COLUMN.'/
               2    ' IF C=Ø END OF CHANGING')
ØØ31       25Ø  WRITE(5,251)
```

```
0032    251  FORMAT('$C: ')
0033         READ(5.252) IC
0034    252  FORMAT(I1)
0035         IF(IC.EQ.0) GOTO 100
0037         IF(IC.GT.8.OR.IC.LT.0) GOTO 250
0039    260  CONTINUE
0040         WRITE(5.261)
0041    261  FORMAT('$R: ')
0042         READ(5.252) IR
0043         IF(IR.EQ.0) GOTO 250
0045         IF(IR.GT.8.OR.IR.LT.0) GOTO 260
0047         WRITE(5.271) IR.IC
0048    271  FORMAT('$AM('.I1.'.'.I1.')= ')
0049         READ(5.272) DAM(IR.IC)
3050    272  FORMAT(F10.0)
0051         GOTO 260
0052    280  CONTINUE
        C
        C    SAM IS THE A-MATRIX OF THE IDEAL MODEL
        C    IN SINGLE DIMENSIONED STORAGE
0053         CALL ARRAY(MOD:N.M.N.M.SAM.DAM)
0054    300  CONTINUE
0055         N=8
0056         M=4
0057         CALL ARRAY(MOD.N.M.N.M.SB.DB)
0058     600 RETURN
0059         END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:13           PAGE 001
CORE=08K. UIC=[123.1]                                          .LP/LI:1=A7MAN

0001            SUBROUTINE FCT(X.Y.DERY)
          C     ------------------------
          C
0002            COMMON /A7/DAT(20).INT(10).R(5)
0003            COMMON/MATR/SA(64).SB(32).SAM(64)
0004            COMMON /CON/IA.A13.A14.A37.SUMP(4)
0005            DIMENSION Y(8).DERY(8).AY(8).BU(8).U(4)
0006            DIMENSION UPILOT(4)
0007            EQUIVALENCE (UPILOT(1).DE).(UPILOT(2).DA).
               1               (UPILOT(3).DR).(UPILOT(4).DT)
0008            EQUIVALENCE(MODE.INT(3)).(W.R(3))
0009            IF(MODE.GT.-2) GOTO 10
          C
          C     THE NEXT TWO STATEMENTS ARE ONLY IMPORTANT.
          C     IF FUNCTIONS FOR DERIVATIVES ARE USED
0011            AL0=19.
0012            BETA0=6.
0013            DO 11 I=1.4
0014            SUMP(I)=0.
0015       11   U(I)=0.
0016            DE=0.
0017            DA=0.
0018            DR=0.
0019            DT=0.
0020            MODE=-1
0021       10   CONTINUE
          C
          C     INT(2) : 'CASE'
          C     CASE 1.2,3 WERE USED TO FIND AND TEST IDEAL MODEL
          C     CASE 4 IS FLIGHT WITH IC-VALUE OF YAW RATE
          C     CASE 5 IS FLIGHT WITH IC-VALUE OF PITCH RATE
0022            GO TO (1.2.3.4.5).INT(2)
0023        1   CONTINUE
0024            IF(X.GT.1.) DE=-.05*(X-1.)
0026            GOTO 100
0027        2   CONTINUE
0028            THC=.01
0029            DE=Y(8)-THC
0030            GOTO 100
0031        3   CONTINUE
0032            THC=.01
0033            DE=(Y(8)-THC)
0034            DT=-Y(1)/10.
0035        4   CONTINUE
0036        5   CONTINUE
0037      100   CONTINUE
0038            DO 150 I=1.4
          C
          C     W: WEIGHTING (SEE MAIN PROGRAM)
0039      150   U(I)=W*SUMP(I)+(1.-W)*UPILOT(I)
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:13          PAGE 002
CORE=08K. UIC=[123.1]                                          ,LP/LI:1=A7MAN

0040          IF(INT(5).NE.1) GOTO 200
      C
      C       THE FOLLOWING PART WAS ONLY USED TO FIND AN
      C       IDEAL MODEL: IT IS NOT USED IN THIS VERSION
      C       OF THE PROGRAM
0042          ALDG=57.295*Y(2)+AL0
0043          BETADG=57.295*Y(4)+BETA0
0044          RLBETA=-8.+.89*(ALDG-14.)
0045          RNBETA=1.2-.267*(ALDG-14.)
0046          ALDGM=ALDG-18.5
0047          RLAL=(5.-.247*ALDGM*ALDGM)*BETADG/8.
0048          RNAL=-.25*BETADG
0049          IF(ALDG.GT.21.) RNAL=RNAL*(1.-(ALDG-21.)/3.)
0051          SA(13)=RLAL
0052          SA(14)=RNAL
0053          SA(29)=RLBETA
0054    200   CONTINUE
      C
      C       OUTPUT PREPARATION
0055          DO 101 I=1.3
0056    101   DAT(10+I)=57.295*U(I)
0057          DAT(14)=U(4)
      C
      C       FORMULATION OF FLIGHT DYNAMIC EQUATIONS
      C
      C            DERIVATIVE OF X=A*X+B*U
      C
      C       IN THIS PROGRAM Y IS USED INSTEAD OF X
      C       (COMPARE PAGE 2 OF TEXT)
      C
      C       IN THE CASE OF IDEAL MODEL SIMULATION
      C       THE AM-MATRIX (SAM) IS USED INSTEAD OF THE A-MATRIX (SA)
0058          N=8
0059          M=8
0060          L=1
0061          SA(13)=A13
0062          SA(14)=A14
0063          SA(37)=A37
0064          IF(INT(4).EQ.1.OR.INT(4).EQ.3) CALL GMPRD(SA.Y.AY.N,M.L)
0066          IF(INT(4).EQ.2) CALL GMPRD(SAM.Y.AY.N.M.L)
0068          M=4
0069          CALL GMPRD(SB.U.BU.N.M.L)
0070          M=1
0071          CALL GMADD(AY.BU.DERY.N.M)
0072          RETURN
0073          END
```

```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:37:16          PAGE 001
CORE=08K, UIC=[123,1]                                          ,LP/LI:1=A7MAN

0001            SUBROUTINE OUTP(X,Y,DERY,IHLF,NDIM,PRMT)
          C     ----------------------------------------
          C
0002            COMMON /A7/DAT(20),INT(10),R(5)
0003            COMMON /KAL/XHATK(8),RKG(64),C(8),CT(8),VRIN(8)
0004            COMMON /NOISE/SIGMA(8),YM(8)
0005            COMMON /OUT/YMOUT(8),SVDOUT(8,5)
0006            DIMENSION Y(8),DERY(8),PRMT(5),CY(8),IG(8),VN(8)
0007            EQUIVALENCE (MODE,INT(3)),(NF,INT(4))
0008            IF(IHLF.EQ.IHLF1) GOTO 300
          C
          C     TO COMPILE STATEMENTS WITH A D IN THE FIRST
          C     COLUMN THE SWITCH /DE MUST BE USED IN THE
          C     COMMAND LINE FOR THE FORTRAN COMPILER
          C     OTHERWISE THE STATEMENTS ARE TREATED AS
          C     COMMENT LINES
          D     WRITE(6,451) IHLF,INT(4)
          D 451 FORMAT('0IHLF:',I2,' AT FLIGHT NO:',I1)
0010            IHLF1=IHLF
0011        300 CONTINUE
          C
          C     DATA TRANSFER
          C
          C     NF MEANS FILE NUMBER
          C
          C     NF=1 : VALUES FOR A7  FILE NAME: FOR001.DAT
          C        2 : VALUES FOR MO             FOR002.DAT
          C        3 : VALUES FOR MF             FOR003.DAT
          C
0012            IF(MODE.GT.-1) GOTO 310
0014            REWIND NF
0015            N=0
0016            DELT=0.
0017            XM=R(1)
0018            WRITE(NF) XM
0019            DELTA=XM/500.
0020            MODE=0
0021        310 CONTINUE
0022            IX=X
0023            IF(IX.EQ.IX1) GOTO 450
          D     WRITE(6,400) IX,INT(4)
          D 400 FORMAT('0FLIGHT TIME: ',I2,' SEC AT FLIGHT NO: ',I1)
0025            IX1=IX
0026        450 CONTINUE
0027            IF(X.LT.DELT) RETURN
          C
          C     NORMAL DATA TRANSFER
0029            DAT(20)=X
0030            DAT(1)=Y(1)
0031            DO 302 I=2,8
```

```
FORTRAN IV      V01B-02           THU 26-MAY-77 14:37:16            PAGE 002
CORE=08K. UIC=[123.13                                             ,LP/LI:1=A7MAN

0032    302  DAT(I)=57.295*Y(I)
0033         WRITE(NF) DAT
0034         N=N+1
0035         DELT=DELTA*N
0036         IF(X.LT.R(1)) RETURN
0038         MODE=1
0039         RETURN
0040         END
```

```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:37:17          PAGE 001
CORE=08K, UIC=[123,1]                                         ,LP/LI:1=A7MAN

0001          SUBROUTINE END
      C       ---------------
      C
      C       THIS SUBROUTINE WAS USED TO SAVE THE
      C       TIME OF END OF CALCULATION
0002          COMMON /A7/DAT(20),INT(10)
0003          EQUIVALENCE (NF,INT(4))
0004          DIMENSION ITIME(4)
0005          CALL TIME(ITIME)
0006          WRITE(NF) ITIME
0007          RETURN
0008          END
```

```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:37:18          PAGE 001
CORE=08K. UIC=[123.1]                                      ,LP/LI:1=A7MAN

0001          SUBROUTINE CONTRL
        C     -----------------
        C
0002          COMMON /A7/DAT(20),INT(10).R(5)
0003          COMMON /MATR/SA(64).SB(32).SAM(64).UM(4)
0004          COMMON /KAL/XHATK(8).RKG(64).C(8).CT(8).VRIN(8)
0005          COMMON /K/RBAR(16).S(32).SBT(32).RKBAR(32)
0006          COMMON /CON/IA.A13.A14.A37.SUMP(4).XHI(8.5)
0007          COMMON /OUT/YMOUT(8).SVDOUT(8.5)
0008          DIMENSION UMP(4).UAR(4.5).PROUT(5)
0009          EQUIVALENCE(KEN. INT(6))
0010          IA=1
0011          DO 10 I=1.4
0012          SUMP(I)=0.
0013     10   CONTINUE
0014      1   CONTINUE
        C
        C     INTRODUCING POSSIBLE PARAMETER VECTORS FOR THE
        C     DERIVATIVES EXPECTED TO BE UNCERTAIN.
        C     (COMPARE TEXT PAGE 12)
0015          SA(13)=A13*(1.-.45*(IA-3))
0016          SA(14)=A14*(1.-.45*(IA-3))
0017          SA(37)=A37*(1.-.45*(IA-3))
0018          CALL SUBCON
0019          DO 50 I=1.4
0020          UAR(I.IA)=UM(I)
0021     50   CONTINUE
0022          IA=IA+1
0023          IF(IA.LE.5) GOTO 1
0025          DO 200 I=1.5
0026          IA=I
        C
        C     CALCULATION OF PROBABILITY
0027          CALL PROB(PR)
        C
        C     PREPARING THE OUTPUT
0028          PROUT(I)=PR
0029          DO 300 N=1.4
0030          UM(N)=UAR(N.IA)
0031    300   CONTINUE
0032          CALL SMPY(UM.PR.UMP.4.1.0)
0033          DO 200 N=1.4
        C
        C     THE CONTROL TERM SUMP(N)  IS THE SOLUTION OF
        C     EQUATION 10 PAGE 6 AND IS USED IN SUBROUTINE FCT
0034          SUMP(N)=SUMP(N)+UMP(N)
0035    200   CONTINUE
        C
        C     PROVIDING THE OUTPUT OF TIME. MEASUREMENTS.PROBABILITIES
        C     AND STANDARD DEVIATIONS (FILE: FOR004.DAT)
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:18          PAGE 002
CORE=08K, UIC=[123,1]                                       ,LP/LI:1=A7MAN

0036          T=DAT(20)
0037          WRITE(4) T,YMOUT,PROUT,SVDOUT
0038          KEN=1
0039          RETURN
0040          END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:20          PAGE 001
CORE=08K, UIC=[123.1]                                          ,LP/LI:1=A7MAN

0001           SUBROUTINE PROB(PR)
         C     --------------------
         C
0002           COMMON /A7/ DAT(20),INT(10),R(5)
0003           COMMON /CON/IA.A13.A14.A37.SUMP(4),XHI(8.5)
0004           COMMON /KAL/XHATK(8),RKG(64).C(8).CT(8),VRIN(8)
0005           COMMON /NOISE/SIGMA(8).YM(8)
0006           DIMENSION XHT(8).XHTT(8).TEMP1(8).TEMP2(8).TEMP3(8)
0007           DIMENSION P(5).ARG(5).ARG0(5)
0008           DIMENSION A(5).NA(5)
0009           EQUIVALENCE(KEN.INT(6))
         C
         C     P(I) ARE THE A PRIORI PROBABILITIES
0010           DATA P/5*.2/.HALF/.5/.ARG0/5*0./.DA/5./
0011           IF(IA.GT.1) GOTO 200
0013           SUM=0.
0014           DO 960 I=1.5
0015           NA(I)=0
0016     960   CONTINUE
0017           DO 100 I=1.5
0018           DO 25 N=1.8
0019           XHT(N)=XHI(N.I)
0020      25   CONTINUE
         C
         C     CALCULATION OF PROBABILITIES (PR) AS
         C     MENTIONED IN THE TEXT OF THIS APPENDIX
0021           CALL GMTRA(XHT.XHTT.8.1)
0022           CALL MPRD(XHTT.CT.TEMP1.1.8.0.2.8)
0023           CALL MPRD(TEMP1.VRIN.TEMP2.1.8.0.2.8)
0024           CALL MPRD(C.XHT.TEMP1.8.8.2.0.1)
0025           CALL SMPY(TEMP1.HALF.TEMP1.8.1.0)
0026           CALL GMSUB(YM.TEMP1.TEMP3.8.1)
0027           CALL GMPRD(TEMP2.TEMP3.DIF.1.8.1)
0028           ARG(I)=DIF*R(2)+ARG0(I)
0029           ARG0(I)=ARG(I)
0030           A(I)=ARG(I)
0031     900   CONTINUE
0032           IF(A(I).GE.87.) GOTO 901
0034     100   CONTINUE
0035     200   CONTINUE
0036           SUM=0.
0037           DO 300 J=1.5
0038           AI=A(J)-A(IA)+(NA(J)-NA(IA))*DA
0039           IF(AI.GT.85) GOTO 902
0041           SUM=SUM+P(J)/P(IA)*EXP(AI)
0042     300   CONTINUE
         C
         C     PR IS THE SOLUTION OF EQUATION 16 PAGE 8
         C     IMPORTANT: PR=PR(IA) COMPARE SUBROUTINE PROB
0043           PR=1./SUM
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:20          PAGE 002
CORE=08K. UIC=[123.1]                                        .LP/LI:1=A7MAN

0044    400   CONTINUE
0045          RETURN
0046    901   CONTINUE
0047          A(I)=A(I)-DA
0048          NA(I)=NA(I)+1
0049          GOTO 900
0050    902   CONTINUE
0051          PR=0.
0052          GOTO 400
0053          END
```

```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:37:22          PAGE 001
CORE=08K, UIC=[123,1]                                          .LP/LI:1=A7MAN

0001           SUBROUTINE SUBCON
         C     -----------------
         C
0002           COMMON /A7/DAT(20).INT(10).R(5)
0003           COMMON/MATR/SA(64).SB(32).SAM(64).UM(4)
0004           COMMON/CONTR/MODEC
0005           COMMON/K/RBAR(16).S(32).SBT(32).RKBAR(32).P
0006           DIMENSION PAR(5).P(64).PDOT(64).STOR(8.64)
0007           DIMENSION XHAT(8)
0008           EQUIVALENCE (KEN.INT(6))
0009           EXTERNAL RICCAT.RESULT
0010           IF(KEN.EQ.1) GOTO 400
         C
         C     THE MAIN GOAL OF THIS SUBROUTINE IS TO SOLVE
         C     THE RICCATI EQUATIONS TO GET P FROM EQUATION 12
         C     PAGE 7 (USING SUBROUTINE RKGS AND SUBROUTINE RICCAT)
0012           MODEC=-1
0013           NO=64
0014           PAR(1)=R(4)
0015           PAR(2)=0.
0016           PAR(3)=-R(5)
0017           PAR(4)=.0001
0018           DO 300 I=1.64
0019           P(I)=0.
0020           PDOT(I)=1./64.
0021     300   CONTINUE
0022           CALL RKGS(PAR.P.PDOT.NO.IHALF.RICCAT.RESULT.STOR)
0023           IF(IHALF.GT.10) GOTO 900
0025           GOTO 999
0026     900   WRITE(5.901) IHALF
0027     901   FORMAT('0*** ERROR *** IHALF: '.I2)
0028     999   CONTINUE
0029           CALL KBARUP
0030      400  CONTINUE
0031           CALL KALMAN(XHAT)
0032           CALL GMPRD(RKBAR.XHAT.UM.4.8.1)
0033           RETURN
0034           END
```

```
[illegible]                      THU 26-MAY-77 14:37:23         PAGE 001
[illegible]                                                  ,LP/LI:1=A7MAN

0001           SUBROUTINE RICCAT(TR.P.PDOT)
         C     ----------------------------
         C

0002           COMMON/MATR/SA(64).SB(32).SAM(64)
0003           COMMON/CONTR/MODEC
0004           COMMON /K/RBAR.S.SBT.RKBAR(32)
0005           DIMENSION OP(8).RP(4).C0(8.8).SC0(64)
0006           DIMENSION SBT(32).SC0T(64).SR1(32).SR2(32)
0007           DIMENSION SR3(32).SR4(32).RBAR(16)
0008           DIMENSION SC0A(64).SAMC0(64).CAMCAM(64).S(32)
0009           DIMENSION CAMT(64).Q1(64).QBAR(64)
0010           DIMENSION LSTORB(4).MSTORB(4).
              A  P(64).PDOT(64).
              B  BR(32).BRS(64).AMBRS(64).ABRST(64).
              C  PDOT1(64).PDOT2(64).PB(32).
              D  PBR(32).PBRBT(64).PDOT3(64).
              E  ST(32).STR(32).PDOT4(64).PDOT5(64)
         C
         C     TASK OF THIS SUBROUTINE: SEE CALLING
         C     SUBROUTINE SUBCON
0011           DATA C0/1..7*0..
              2  1*0..1..6*0..
              3  2*0..1..5*0..
              4  3*0..1..4*0..
              5  4*0..1..3*0..
              6  5*0..1..2*0..
              7  6*0..1..1*0..
              8  7*0..1./
0012           DATA C/-1./
0013           IF(MODEC) 1.2.3
0014       1   CONTINUE
0015           OP(1)=1.
0016           OP(2)=1.
0017           OP(3)=1.
0018           OP(4)=1.
0019           OP(5)=1.
0020           OP(6)=1.
0021           OP(7)=1.
0022           OP(8)=1.
0023           RP(1)=1.
0024           RP(2)=1.
0025           RP(3)=1.
0026           RP(4)=0.
0027           CALL ARRAY(2.8.8.8.8.SC0.C0)
0028           CALL GMTRA(SB.SBT.8.4)
0029           CALL GMTRA(SC0.SC0T.8.8)
0030           CALL GMPRD(SBT.SC0T.SR1.4.8.8)
0031           CALL MPRD(SR1.OP.SR2.4.8.0.2.3)
0032           CALL GMPRD(SR2.SC0.SR3.4.8.8)
0033           CALL GMPRD(SR3.SB.SR4.4.8.4)
```

```
FORTRAN
CORE=08

0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
        C
        C
        C
0079
0081
0083
        C
        C
        C
        C
0084
0085
0086
        C
        C
        C
0087
0088
0089
0090
        C
        C
0092
0093
0094
0095
0096
        C
        C
0097
0098
0099
0100
        C
        C
0101
        C
        C
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:23          PAGE 002
CORE=08K. UIC=[123.1]                                           .LP/LI:1=A7MAN

0034            CALL MADD(SR4.RP.RBAR.4.4.0.2)
0035            CALL GMPRD(SC0.SA.SC0A.8.8.8)
0036            CALL GMPRD(SAM.SC0.SAMC0.8.8.8)
0037            CALL GMSUB(SC0A.SAMC0.CAMCAM.8.8)
0038            CALL GMPRD(SR2.CAMCAM.S.4.8.8)
0039            CALL GMTRA(CAMCAM.CAMT.8.8)
0040            CALL MPRD(CAMT.QP.Q1.8.8.0.2.8)
0041            CALL GMPRD(Q1.CAMCAM.QBAR.8.8.8)
0042            CALL MINV(RBAR.4.DETB.LSTORB.MSTORB)
0043            CALL GMPRD(SB.RBAR.BR.8.4.4)
0044            CALL GMPRD(BR.S.BRS.8.4.8)
0045            CALL GMSUB(SA.BRS.AMBRS.8.8)
0046            CALL GMTRA(S.ST.4.8)
0047            CALL GMPRD(ST.RBAR.STR.8.4.4)
0048            CALL GMPRD(STR S.PDOT4.8.4.8)
0049            MODEC=0
0050      2     CONTINUE
0051            CALL GMPRD(P.AMBRS.PDOT1.8.8.8)
0052            CALL GMTRA(AMBRS.ABRST.8.8)
0053            CALL GMPRD(ABRST.P.PDOT2.8.8.8)
0054            CALL GMPRD(P.SB.PB.8.8.4)
0055            CALL GMPRD(PB.RBAR.PBR.8.4.4)
0056            CALL GMPRD(PBR.SBT.PBRBT.8.4.8)
0057            CALL GMPRD(PBRBT.P.PDOT3.8.8.8)
0058            CALL GMADD(PDOT1.PDOT2.PDOT5.8.8)
0059            CALL GMSUB(PDOT5.PDOT3.PDOT1.8.8)
0060            CALL GMADD(PDOT1.QBAR.PDOT5.8.8)
0061            CALL GMSUB(PDOT5.PDOT4.PDOT.8.8)
0062            CALL SMPY(PDOT.C.PDOT.8.8.0)
0063      3     CONTINUE
0064            RETURN
0065            END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:27          PAGE 001
CORE=08K, UIC=[123,1]                                          ,LP/LI:1=A7MAN

0001           SUBROUTINE RESULT(TR,P,PDOT,IHALF,NO,PAR)
         C     -----------------------------------------
         C
0002           COMMON /A7/DAT(20),INT(10),R(5)
0003           DIMENSION P(64),PDOT(64),PAR(5)
         C
       . C     CONCERNING D IN FIRST COLUMN SEE
         C     COMMENT IN SUBROUTINE OUTP
0004           ITR=TR+.999
0005           IF(ITR.EQ.ITR1) GOTO 5
         D     WRITE(6,6) ITR
         D   6 FORMAT(' INT.-TIME RICCATI:',I2)
0007           ITR1=ITR
0008       5   CONTINUE
0009           IF(IHALF.EQ.IHALF1) GOTO 600
         D     WRITE(6,200) IHALF
         D 200 FORMAT('0IHALF:',I2)
0011           IHALF1=IHALF
0012     600   RETURN
0013           END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:28            PAGE 001
CORE=08K. UIC=[123.1]                                          .LP/LI:1=A7MAN

0001           SUBROUTINE KBARUP
       C       -----------------
       C
0002           REAL K1
0003           COMMON/K/RBAR(16),S(32).SBT(32).RKBAR(32).P(64)
0004           DIMENSION SBTP(32).K1(32)
       C
       C       THIS SUBROUTINE SOLVES EQUATION 11 PAGE 7
0005           DATA C/-1./
0006           CALL GMPRD(SBT.P.SBTP.4.8.8)
0007           CALL GMADD(S.SBTP.K1.4.8)
0008           CALL GMPRD(RBAR.K1.RKBAR.4.4.8)
0009           CALL SMPY(RKBAR.C.RKBAR.4.8.0)
0010           RETURN
0011           END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:29          PAGE 001
CORE=08K, UIC=[123,1]                                        ,LP/LI:1=A7MAN

0001          SUBROUTINE KALMAN(XHAT)
        C     -----------------------
        C
0002          COMMON /A7/DAT(20),INT(10),R(5)
0003          COMMON /MATR/ SA(64),SB(32)
0004          COMMON /KAL/ XHATK(8),RKG(64),C(8),CT(8).VRIN(8)
0005          COMMON /NOISE/SIGMA(8).YM(8)
0006          COMMON /K/ RBAR(16).S(32).SBT(32).RKBAR(32)
0007          COMMON /OUT/YMOUT(8),SVDOUT(8.5)
0008          COMMON /CON/IA
0009          DIMENSION XHAT(8)
0010          EQUIVALENCE(KEN.INT(6))
        C
        C     THE TASK OF THIS SUBROUTINE IS TO SOLVE THE
        C     FILTER EQUATIONS (EQUATION 13 PAGE 7)
        C     TO GET XHAT
0011          DATA AM/0./
0012          IF(KEN.EQ.1) GOTO 400
0014          IF(LKEN.EQ.1) GOTO 400
0016          LKEN=1
0017          REWIND 4
0018          WRITE(4) R(2)
0019          DO 200 I=1.8
0020          VRIN(I)=1./(SIGMA(I)*SIGMA(I))
0021      200 CONTINUE
0022          CALL MTRA(C,CT.8.8.2)
0023      400 CONTINUE
        C
        C     CALCULATE KG (EQUATION 14 PAGE 7)
0024          CALL KGSUB
        C
        C     CALCULATE XHAT (EQUATION 13 PAGE 7)
0025          CALL XHATSB
0026          DO 300 I=1.8
0027          XHAT(I)=XHATK(I)
0028      300 CONTINUE
0029          RETURN
0030          END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:30          PAGE 001
CORE=08K, UIC=[123,1]                                          .LP/LI:1=A7MAN

0001          SUBROUTINE KGSUB
        C     ----------------
        C
0002          COMMON /A7/DAT(20),INT(10).R(5)
0003          COMMON /KAL/ XHATK(8).RKG(64).C(8),CT(8).VRIN(8)
0004          COMMON /CON/IA
0005          COMMON /OUT/YMOUT(8),SVDOUT(8.5)
0006          DIMENSION V(64).STORV(8.64).PT(5).VDOT(64).VCT(64),SIG(8)
0007          DIMENSION VLOCAL(64.5)
0008          EXTERNAL VCAL.VRES
0009          EQUIVALENCE(KEN.INT(6))
0010          DATA NO/64/
        C
        C     WITH SUBROUTINE RKGS AND SUBROUTINE VCAL
        C     EQUATION 15 PAGE 8 IS SOLVED TO GET V
0011          IF(KEN.EQ.1) GOTO 400
0013          PT(3)=R(2)/2.
0014          PT(4)=.001
0015          SIG(1)=10.
0016          SIG(2)=.1
0017          SIG(3)=.2
0018          SIG(4)=.1
0019          SIG(5)=.2
0020          SIG(6)=.2
0021          SIG(7)=.1
0022          SIG(8)=.1
0023          DO 100 I=1,64
0024          VLOCAL(I.IA)=0.
0025      100 CONTINUE
0026          DO 200 I=1.8
0027          VLOCAL(8*(I-1)+I.IA)=SIG(I)**2
0028      200 CONTINUE
0029      400 CONTINUE
0030          IF(IA.EQ.1) PT(1)=PT(2)
0032          PT(2)=PT(1)+R(2)
0033          DO 450 I=1.64
0034          VDOT(I)=1./64.
0035          V(I)=VLOCAL(I.IA)
0036      450 CONTINUE
0037          SVDOUT(1.IA)=SQRT(V(1))
0038          DO 300 I=2,8
0039          SVDOUT(I.IA)=57.295*SQRT(V(8*(I-1)+I))
0040      300 CONTINUE
0041          CALL RKGS(PT.V.VDOT.NO.IHFF.VCAL.VRES.STORV)
0042          DO 500 I=1.64
0043     500  VLOCAL(I.IA)=V(I)
        C
        C     CALCULATION OF KG
0044          CALL MPRD(V.CT.VCT.8,8.0.2,8)
0045          CALL MPRD(VCT.VRIN.RKG,8.8,0.2,8)
```

```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:37:30          PAGE 002
CORE=08K. UIC=[123.1]                                    ,LP/LI:1=A7MAN

0046          RETURN
0047          END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:32           PAGE 001
CORE=08K, UIC=[123.1]                                          .LP/LI:1=A7MAN

0001             SUBROUTINE VCAL(X.V.VDOT)
           C     -------------------------
           C
0002             COMMON /MATR/ SA(64)
0003             COMMON /KAL/ XHATK(8).RKG(64).C(8).CT(8).VRIN(8)
0004             DIMENSION V(64).VDOT(64).TEMP1(64).TEMP2(64).TEMP3(64)
0005             DIMENSION AT(64)
           C
           C     SOLVING EQUATION 15 PAGE 8 TO GET V
0006             DO 100 I=1.8
0007             IDIAG=8*(I-1)+I
0008             IF(V(IDIAG).LT.0.) V(IDIAG)=0.
0010         100 CONTINUE
0011             CALL GMTRA(SA.AT.8.8)
0012             CALL GMPRD(SA.V.TEMP1.8.8.8)
0013             CALL GMPRD(V.AT.TEMP2.8.8.8)
0014             CALL GMADD(TEMP1.TEMP2.TEMP3.8.8)
0015             CALL MPRD(V.CT.TEMP1.8.8.0.2.8)
0016             CALL MPRD(TEMP1.VRIN.TEMP2.8.8.0.2,8)
0017             CALL MPRD(TEMP2.C.TEMP1.8.8.0.2.8)
0018             CALL GMPRD(TEMP1.V.TEMP2.8.8.8)
0019             CALL GMSUB(TEMP3.TEMP2.VDOT.8.8)
0020             RETURN
0021             END
```

```
FORTRAN IV      V01B-02        THU 26-MAY-77 14:37:33         PAGE 001
CORE=08K, UIC=[123.1]                                       .LP/LI:1=A7MAN

0001          SUBROUTINE VRES(X.V.VDOT.IHFF.PT)
        C     --------------------------------
        C
0002          DIMENSION V(64).VDOT(64).PT(5)
        C
        C     CONCERNING D IN FIRST COLUMN SEE
        C     COMMENT IN SUBROUTINE OUTP
0003          IF(IHFF.EQ.IHFF1) GOTO 1
        D     WRITE(6.100) IHFF
        D 100 FORMAT('0IHFF IN KALMAN (V):',I2)
0005          IHFF1=IHFF
0006        1 CONTINUE
0007          RETURN
0008          END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:34          PAGE 001
CORE=08K, UIC=[123,1]                                        ,LP/LI:1=A7MAN

0001          SUBROUTINE XHATSB
      C       -----------------
      C
0002          COMMON /A7/ DAT(20),INT(10),R(5)
0003          COMMON /KAL/ XHATK(8),RKG(64),C(8),CT(8),VRIN(8)
0004          COMMON/K/RBAR(16),S(32),SBT(32),RKBAR(32)
0005          COMMON /CON/IA
0006          DIMENSION XH(8),XHDOT(8),STORXH(8,8),PXH(5)
0007          DIMENSION XHLOC(8,5)
0008          EXTERNAL XHCAL,XHRES
0009          EQUIVALENCE (KEN,INT(6))
      C
      C       WITH SUBROUTINE RKGS AND SUBROUTINE XHCAL
      C       THE FILTER EQUATIONS APE SOLVED
0010          DATA NOX/8/
0011          IF(KEN.EQ.1) GOTO 400
0013          PXH(3)=R(2)/2.
0014          PXH(4)=.001
0015          DO 100 I=1,8
0016          XHLOC(I,IA)=0.
0017      100 CONTINUE
0018          IF(INT(2).EQ.4) XHLOC(6,IA)=-10./57.295
0020      400 CONTINUE
0021          IF(IA.EQ.1) PXH(1)=PXH(2)
0023          PXH(2)=PXH(1)+R(2)
0024          DO 450 I=1,8
0025          XHDOT(I)=1./8.
0026          XH(I)=XHLOC(I,IA)
0027      450 CONTINUE
0028          CALL RKGS(PXH,XH,XHDOT,NOX,IHXH,XHCAL,XHRES,STORXH)
0029          DO 200 I=1,8
0030          XHATK(I)=XH(I)
0031          XHLOC(I,IA)=XH(I)
0032      200 CONTINUE
0033          RETURN
0034          END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:37:35          PAGE 001
CORE=08K, UIC=[123,1]                                      ,LP/LI:1=A7MAN

0001          SUBROUTINE XHCAL(X,XH,XHDOT)
      C       ---------------------------
      C
0002          COMMON /A7/DAT(20).INT(10).R(5)
0003          COMMON /MATR/SA(64).SB(32)
0004          COMMON /KAL/ XHATK(8).RKG(64).C(8).CT(8).VRIN(8)
0005          COMMON /NOISE/SIGMA(8).YM(8)
0006          COMMON/K/RBAR(16).S(32).SBT(32).RKBAR(32)
0007          DIMENSION XH(8).XHDOT(8).TEMP1(8).TEMP2(64)
0008          DIMENSION TEMP3(8).TEMP4(8).TEMP5(8).YML(8)
      C
      C       EQUATION 13 PAGE 7 IS USED TO GET XHAT
0009          CALL GMPRD(SA.XH.TEMP1.8.8.1)
0010          CALL GMPRD(SB.RKBAR.TEMP2.8.4.8)
0011          CALL GMPRD(TEMP2.XH.TEMP3.8.8.1)
0012          CALL MPRD(C.XH.TEMP4.8.8.2.0.1)
0013          CALL GMSUB(YM.TEMP4.TEMP5.8.1)
0014          CALL GMPRD(RKG.TEMP5.TEMP4.8.8.1)
0015          CALL GMADD(TEMP1.TEMP3.TEMP5.8.1)
0016          CALL GMADD(TEMP5.TEMP4.XHDOT.8.1)
0017          RETURN
0018          END
```

```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:37:37          PAGE 001
CORE=08K, UIC=[123,1]                                        ,LP/LI:1=A7MAN

0001          SUBROUTINE XHRES(X,XH,XHDOT,IHXH,NXH,PXH)
        C     ------------------------------------------
        C
0002          COMMON/A7/ DAT(20),INT(10),R(5)
0003          COMMON /CON/IA,A13,A14,A37,SUMP(4),XHI(8,5)
0004          DIMENSION XH(8),XHDOT(8),PXH(5)
0005          EQUIVALENCE (INT(7),IHMAX)
        C
        C     THIS SUBROUTINE PROVIDES THE MAXIMUM NUMBER
        C     OF BISECTIONS OF THE INITIAL TIME INCREMENT
        C     IN THE XHAT-CALCULATION
0006          IHMAX=MAX0(IHMAX,IHXH)
0007          DO 200 I=1,8
0008          XHI(I,IA)=XH(I)
0009      200 CONTINUE
0010          RETURN
0011          END
```

APPENDIX C
Model Following Program

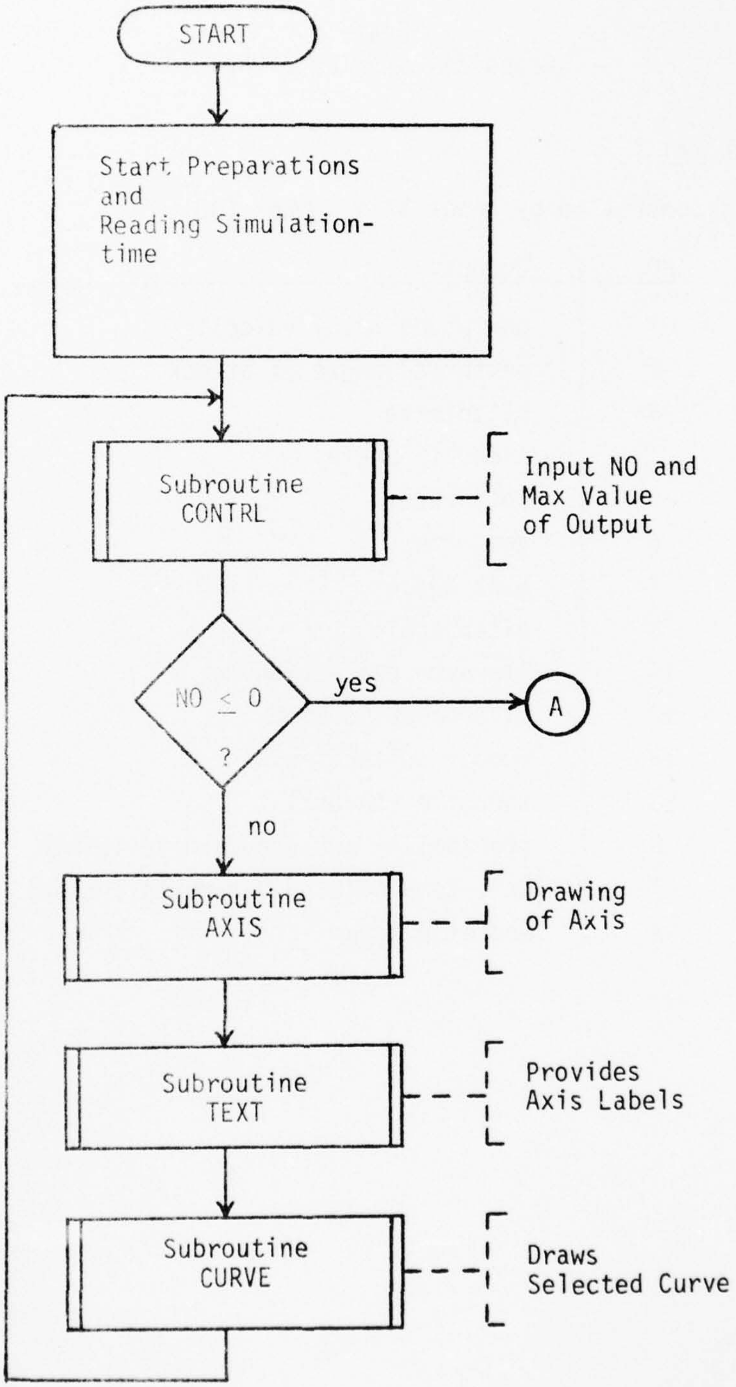
-Part 2-
Graphical Display of Results

1. Program Control

The program was controlled by input of a number (NO):

NO	Output
1	perturbed total velocity
2	perturbed angle of attack
3	pitch rate
4	sideslip angle
5	roll rate
6	yaw rate
7	bank angle
8	pitch angle
11	elevator deflection
12	aileron deflection
13	rudder deflection
14	throttle (thrust)
0	probability and standard deviation
-1	as 1 to 8 with measurements (noise)
-2	end of program

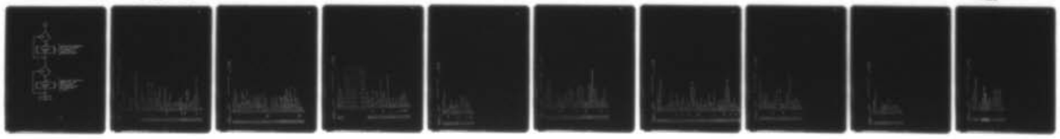
2. Flow Chart Program, Part 2



AD-A041 436 FRANK J SEILER RESEARCH LAB UNITED STATES AIR FORCE --ETC F/G 19/5
HIGH ANGLE OF ATTACK FLIGHT CONTROL USING STOCHASTIC MODEL REFE--ETC(U)
MAY 77 R B ASHER, D GOEBEL

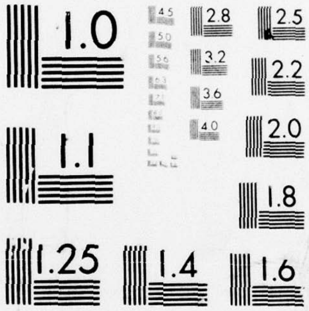
UNCLASSIFIED FJSRL-TR-77-0010 NL

2 OF 2
AD
A041436

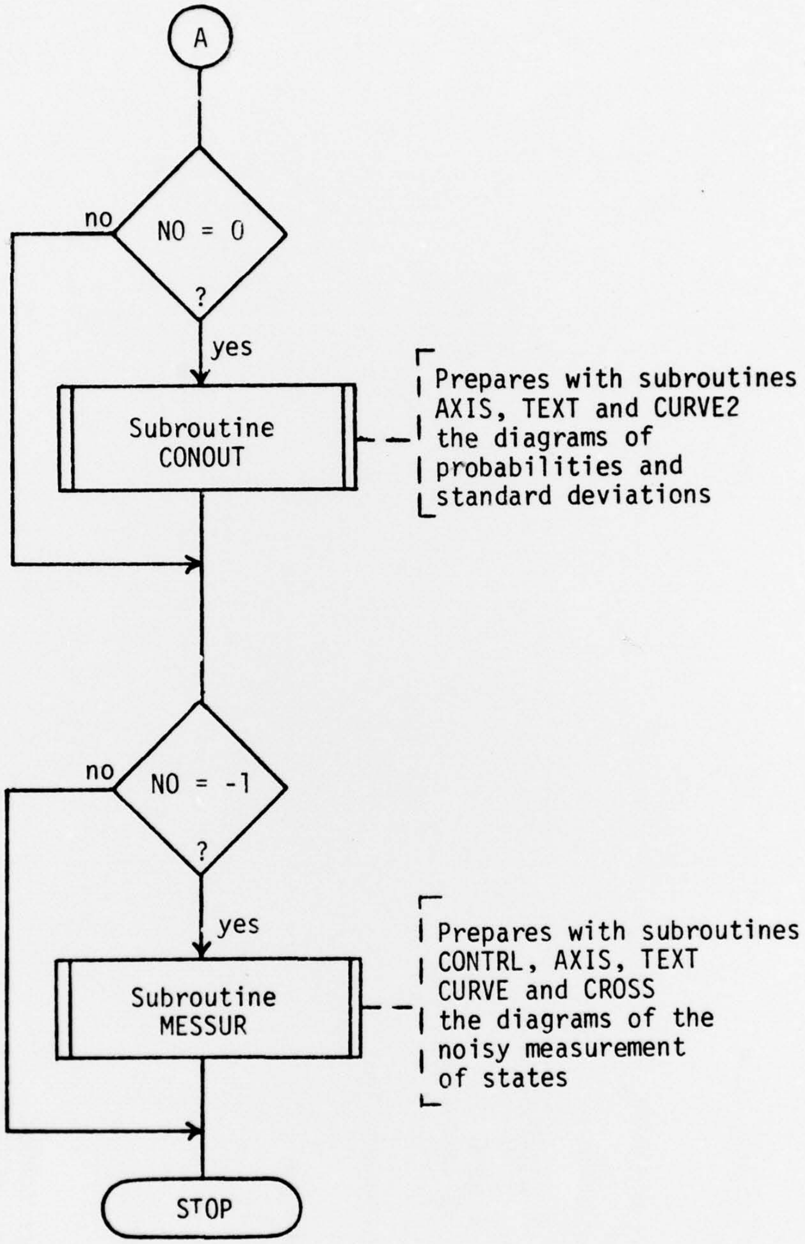


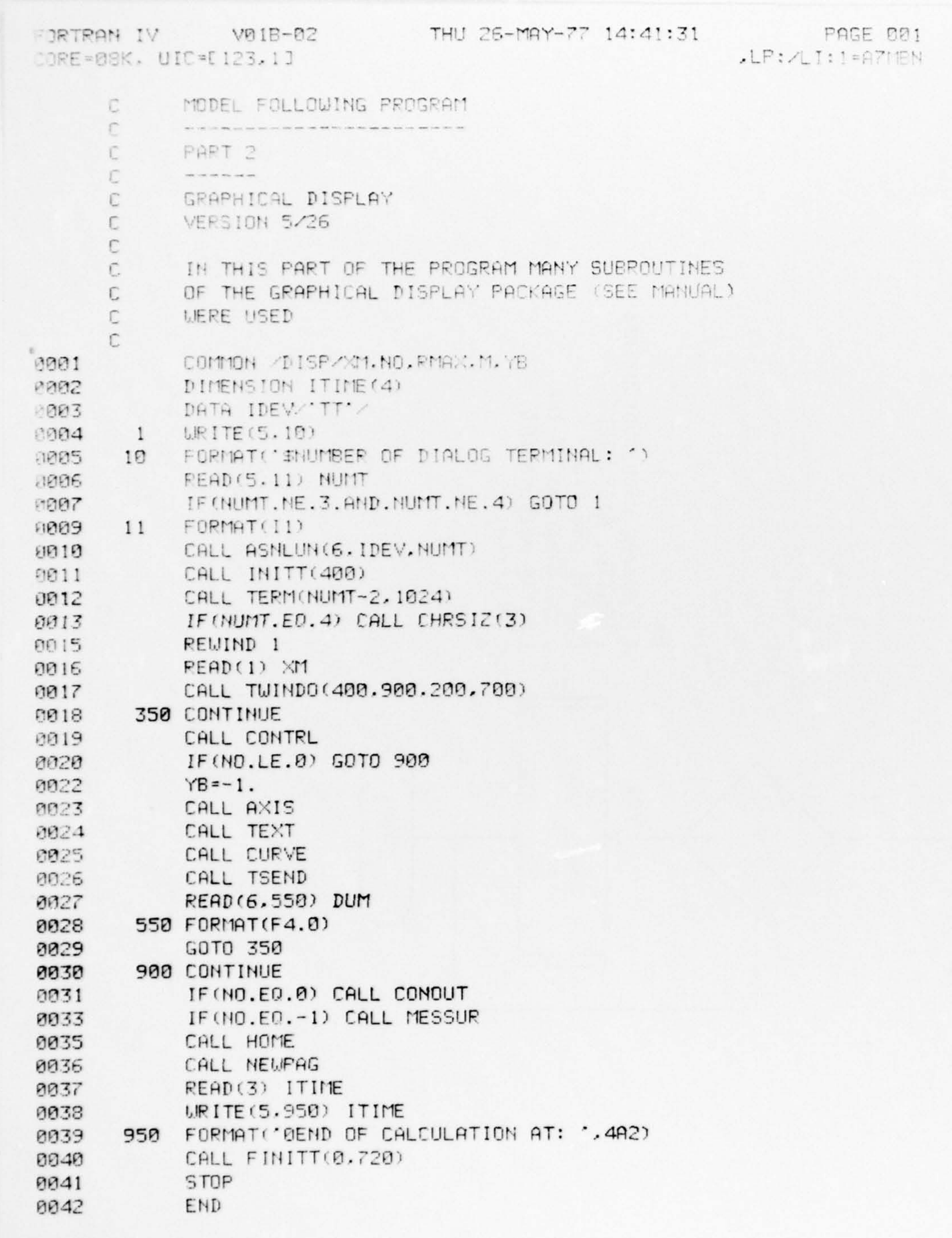
END

DATE
FILMED
7 - 77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A


A
no
NO = 0
?
yes
Subroutine CONOUT
Prepares with subroutines AXIS, TEXT and CURVE2 the diagrams of probabilities and standard deviations
no
NO = -1
?
yes
Subroutine MESSUR
Prepares with subroutines CONTRL, AXIS, TEXT CURVE and CROSS the diagrams of the noisy measurement of states
STOP



```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:41:31          PAGE 001
CORE=08K. UIC=[123,1]                                     ,LP:/LI:1=A7MBN

      C      MODEL FOLLOWING PROGRAM
      C      -----------------------
      C      PART 2
      C      ------
      C      GRAPHICAL DISPLAY
      C      VERSION 5/26
      C
      C      IN THIS PART OF THE PROGRAM MANY SUBROUTINES
      C      OF THE GRAPHICAL DISPLAY PACKAGE (SEE MANUAL)
      C      WERE USED
      C
0001         COMMON /DISP/XM.NO.RMAX.M.YB
0002         DIMENSION ITIME(4)
0003         DATA IDEV/'TT'/
0004     1   WRITE(5.10)
0005    10   FORMAT('$NUMBER OF DIALOG TERMINAL: ')
0006         READ(5.11) NUMT
0007         IF(NUMT.NE.3.AND.NUMT.NE.4) GOTO 1
0009    11   FORMAT(I1)
0010         CALL ASNLUN(6.IDEV,NUMT)
0011         CALL INITT(400)
0012         CALL TERM(NUMT-2,1024)
0013         IF(NUMT.EQ.4) CALL CHRSIZ(3)
0015         REWIND 1
0016         READ(1) XM
0017         CALL TWINDO(400.900.200,700)
0018     350 CONTINUE
0019         CALL CONTRL
0020         IF(NO.LE.0) GOTO 900
0022         YB=-1.
0023         CALL AXIS
0024         CALL TEXT
0025         CALL CURVE
0026         CALL TSEND
0027         READ(6.550) DUM
0028     550 FORMAT(F4.0)
0029         GOTO 350
0030     900 CONTINUE
0031         IF(NO.EQ.0) CALL CONOUT
0033         IF(NO.EQ.-1) CALL MESSUR
0035         CALL HOME
0036         CALL NEWPAG
0037         READ(3) ITIME
0038         WRITE(5.950) ITIME
0039     950 FORMAT('0END OF CALCULATION AT: ',4A2)
0040         CALL FINITT(0.720)
0041         STOP
0042         END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:42:02           PAGE 001
CORE=08K, UIC=[123.1]                                        ,LP:/LI:1=A7MBN

              SUBROUTINE AXIS
              COMMON /DISP/XM,NO.RMAX.M.YB
              CALL DWINDO(0..XM.YB.1.)
              CALL NEWPAG
              IDELTA=100
              IF(XM.LE.1000.) IDELTA=10
              IF(XM.LE.100.) IDELTA=1
              IXM=XM+IDELTA
              DO 351 IX=IDELTA.IXM,IDELTA
              XPLOT=IX-IDELTA
              CALL MOVEA(XPLOT,YB)
              YL=YB+.02
              IF(IX.EQ.IDELTA) YL=1.
              IF(IX.EQ.IXM) YL=1.
              CALL DRAWA(XPLOT,YL)
              IXP=400+500*XPLOT/XM
              CALL MOVABS(IXP,175)
              CALL TSEND
              XOUT=XM*(IX-1)/(IXM-1)
              WRITE(6,500) XOUT
        500   FORMAT('+',F2.0)
        351   CONTINUE
              NY=1
              IF(YB.GE.0.) NY=11
              DO 352 IY=NY,21
              YPLOT=(IY-11)/10.
              CALL MOVEA(0.,YPLOT)
              XL=.01*XM
              IF(IY.EQ.11) XL=XM
              IF(IY.EQ.1) XL=XM
              IF(IY.EQ.21) XL=XM
              CALL DRAWA(XL,YPLOT)
              IF(MOD(IY,2).EQ.0) GOTO 352
              IYP=200+25*(IY-1)
              IF(YB.GE.0) IYP=200+50*(IY-11)
              CALL MOVABS(300.IYP)
              CALL TSEND
              YOUT=-RMAX+2*RMAX*(IY-1)/20
              WRITE(6,501) YOUT
        501   FORMAT('+',F6.1)
        352   CONTINUE
              RETURN
              END
```

```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:42:10          PAGE 001
CORE=08K. UIC=[123.1]                                      .LP:/LI:1=A7MBN

0001            SUBROUTINE TEXT
0002            COMMON /DISP/XM.NO.RMAX.M.YB
0003            DIMENSION ITEXT1(10.14)
0004            DATA ITEXT1/
               1'VE'.'LO'.'CI'.'TY'.'  '.'  '.'  '.'  '.'  '.'  '.
               2'AN'.'GL'.'E '.'OF'.' A'.'TT'.'AC'.'K '.'  '.'  '.
               3'PI'.'TC'.'H '.'RA'.'TE'.'  '.'  '.'  '.'  '.'  '.
               4'SI'.'DE'.' S'.'LI'.'P '.'AN'.'GL'.'E '.'  '.'  '.
               5'RO'.'LL'.' R'.'AT'.'E '.'  '.'  '.'  '.'  '.'  '.
               6'YA'.'W '.'RA'.'TE'.'  '.'  '.'  '.'  '.'  '.'  '.
               7'BA'.'NK'.' A'.'NG'.'LE'.'  '.'  '.'  '.'  '.'  '.
               8'PI'.'TC'.'H '.'AN'.'GL'.'E '.'  '.'  '.'  '.'  '.
               9'  '.'  '.'  '.'  '.'  '.'  '.'  '.'  '.'  '.'  '.
               A'  '.'  '.'  '.'  '.'  '.'  '.'  '.'  '.'  '.'  '.
               1'EL'.'EV'.'AT'.'OR'.' D'.'EF'.'LE'.'CT'.'IO'.'N '.
               2'AI'.'LE'.'RO'.'N '.'DE'.'FL'.'EC'.'TI'.'ON'.'  '.
               3'RU'.'DD'.'ER'.' D'.'EF'.'LE'.'CT'.'IO'.'N '.'  '.
               4'TH'.'RO'.'TT'.'LE'.'  '.'  '.'  '.'  '.'  '.'  '/
0005            CALL MOVABS(680.150)
0006            CALL TSEND
0007            WRITE(6.100)
0008      100   FORMAT('+TIME   [SEC]')
0009            IXT=330
0010            IF(YB.GE.0) IXT=510
0012            CALL MOVABS(IXT.760)
0013            CALL TSEND
0014            WRITE(6.200) (ITEXT1(I.NO).I=1.10)
0015      200   FORMAT('+'.10A2)
0016            CALL MOVABS(330.730)
0017            CALL TSEND
0018            GO TO (1.2.3.2.3.3.2.2.5.5.2.2.2.4).NO
0019      1     WRITE(6.201)
0020            GOTO 5
0021      2     WRITE(6.202)
0022            GOTO 5
0023      3     WRITE(6.203)
0024            GOTO 5
0025      4     WRITE(6.204)
0026      201   FORMAT('+[FPS]')
0027      202   FORMAT('+[DEG]')
0028      203   FORMAT('+[DEG/SEC]')
0029      204   FORMAT('+[ - ]')
0030      5     RETURN
0031            END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:42:18          PAGE 001
CORE=08K, UIC=[123.1]                                          ,LP:/LI:1=A7MBN

0001            SUBROUTINE CONTRL
0002            COMMON /DISP/XM.NO.RMAX.M.YB
0003            IYW=700
0004      500   CONTINUE
0005            IYW=IYW-100
0006            CALL MOVABS(0.IYW)
0007            CALL TSEND
0008            WRITE(6.501)
0009      501   FORMAT('$NO: ')
0010            READ(6.502) NO
0011      502   FORMAT(I2)
0012            IF(NO.GT.20) GOTO 500
0014            IF(NO.LE.0) GOTO 600
0016            WRITE(6.511)
0017     511    FORMAT('$MAX VALUE: ')
0018            READ(6.512) RMAX
0019     512    FORMAT(F6.0)
0020     600    RETURN
0021            END
```

```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:42:23          PAGE 001
CORE=08K, UIC=[123,1]                                          ,LP:/LI:1=A7MBN

           SUBROUTINE CURVE
           COMMON /DISP/XM,NO,RMAX,M,YB
           DIMENSION DAT(20),NUM(3),ITIME(4)
           DATA NUM/'A7','MO','MF'/
           DY0=.04
           DX=.02*XM
           NFILE=1
           IF(M.EQ.1) NFILE=3
     100   REWIND NFILE
           READ(NFILE) XM
      400  CONTINUE
           READ(NFILE) DAT
           YPLOT=DAT(NO)/RMAX
           XPLOT=DAT(20)
           IF(XPLOT.GT.0.) GOTO 401
           CALL MOVEA(0.,YPLOT)
           XNUM=XM*(.5+NFILE/10.)
           GOTO 400
     401   CONTINUE
           CALL DRAWA(XPLOT,YPLOT)
           IF(XPLOT.GT.XNUM) GOTO 450
           YPLOT1=YPLOT
           IF(XPLOT.GE.XM) GOTO 800
           GOTO 400
     450   CONTINUE
           DY=DY0
           IF(YPLOT.GT.YPLOT1) DY=-DY
           CALL DRAWA(XPLOT+DX,YPLOT+DY)
           IF(DY.LT.0.) CALL MOVEA(XPLOT+DX,YPLOT+3.*DY)
           CALL AOUTST(2,NUM(NFILE))
           CALL MOVEA(XPLOT,YPLOT)
           XNUM=2.*XM
           GOTO 400
     800   CONTINUE
           NFILE=NFILE+1
           IF(NFILE.LE.3) GOTO 100
           RETURN
           END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:42:37           PAGE 001
CORE=08K, UIC=[123.1]                                        ,LP:/LI:1=A7MBN

0001          SUBROUTINE CONOUT
0002          COMMON /DISP/XM.NO.RMAX.M.YB
0003          DIMENSION YMOUT(8).PROUT(5).SVDOUT(8.5)
0004          YB=0.
0005          XMAX=XM
0006          RMAX=1.
0007          YMAX=RMAX
0008          CALL AXIS
0009          CALL MOVABS(680.150)
0010          CALL TSEND
0011          WRITE(6.50)
0012    50    FORMAT('+TIME   [SEC]')
0013          CALL MOVABS(330.730)
0014          CALL TSEND
0015          WRITE(6.60)
0016    60    FORMAT('+PROBABILITY')
0017          DO 100 I=1.5
0018          REWIND 4
0019          READ(4) DELTIM
0020    500   CONTINUE
0021          READ(4) T.YMOUT.PROUT.SVDOUT
0022          CALL CURVE2(T.PROUT(I).I.XMAX.YMAX)
0023          IF(T.LT.XM-DELTIM) GOTO 500
0025          CALL TSEND
0026    550   FORMAT(F4.0)
0027    100   CONTINUE
C028          READ(6.550) DUM
0029          RMAX=10.
0030          YMAX=RMAX
0031          DO 200 I=1.8
0032          CALL AXIS
0033          CALL MOVABS(330.760)
0034          CALL TSEND
0035          WRITE(6.70)
0036    70    FORMAT('+STD. DEV. OF ')
0037          NO=I
0038          CALL TEXT
0039          DO 300 N=1.5
0040          REWIND 4
0041          READ(4) DELTIM
0042    400   CONTINUE
0043          READ(4) T.YMOUT.PROUT.SVDOUT
0044          CALL CURVE2(T.SVDOUT(I.N).N.XMAX.YMAX)
0045          IF(T.LT.XM-DELTIM) GOTO 400
0047    300   CONTINUE
0048          CALL TSEND
0049          READ(6.550) DUM
0050    200   CONTINUE
0051          RETURN
0052          END
```

```
FORTRAN IV      V01B-02         THU 26-MAY-77 14:42:47          PAGE 001
CORE=08K. UIC=[123.1]                                        .LP:/LI:1=A7MBN

0001            SUBROUTINE CURVE2(X.Y.N.XMAX.YMAX)
0002            Y=Y/YMAX
0003            IF(X.GT.0.) GOTO 401
0005            CALL MOVEA(0..Y)
0006            XNUM=XMAX*(.3+N/15.)
0007            DY0=.04
0008            DX=.02*XMAX
0009            RETURN
0010      401   CONTINUE
0011            CALL DRAWA(X.Y)
0012            IF(X.GT.XNUM) GOTO 450
0014            Y1=Y
0015            IF(X.GE.XMAX) CALL TSEND
0017            RETURN
0018      450   CONTINUE
0019            DY=DY0
0020            IF(Y.GT.Y1) DY=-DY
0022            CALL DRAWA(X+DX.Y+DY)
0023            IF(DY.LT.0.) CALL MOVEA(X+DX.Y+2.*DY)
0025            CALL TSEND
0026            WRITE(6.1) N
0027      1     FORMAT('+'.I1)
0028            CALL MOVEA(X.Y)
0029            XNUM=2.*XMAX
0030            RETURN
0031            END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:42:52          PAGE 001
CORE=08K, UIC=[123,1]                                        .LP:/LI:1=A7MBN

0001          SUBROUTINE MESSUR
0002          COMMON /DISP/XM.NO,RMAX.M.YB
0003          YB=-1.
0004          M=1
0005     1    CONTINUE
0006          READ(6.2) DUM
0007     2    FORMAT(F4.0)
0008          CALL NEWPAG
0009          CALL CONTRL
0010          IF(NO.LE.0) RETURN
0012          CALL AXIS
0013          CALL TEXT
0014          CALL CURVE
0015          CALL CROSS
0016          CALL TSEND
0017          GOTO 1
0018          RETURN
0019          END
```

```
FORTRAN IV      V01B-02          THU 26-MAY-77 14:42:59          PAGE 001
CORE=08K, UIC=[123.1]                                        ,LP:/LI:1=A7MBN

0001          SUBROUTINE CROSS
0002          COMMON /DISP/XM.NO,RMAX.M
0003          DIMENSION YMOUT(8).PROUT(5).SVDOUT(8.5)
0004          DY=.01
0005          DX=XM/200.
0006          REWIND 4
0007          READ(4) DELTIM
0008     1    READ(4) T.YMOUT.PROUT.SVDOUT
0009          YPLOT=YMOUT(NO)/RMAX
0010          XPLOT0=T-DX
0011          YPLOT0=YPLOT-DY
0012          XPLOT1=T+DX
0013          YPLOT1=YPLOT+DY
0014          CALL MOVEA(XPLOT0.YPLOT0)
0015          CALL DRAWA(XPLOT1.YPLOT1)
0016          CALL MOVEA(XPLOT0.YPLOT1)
0017          CALL DRAWA(XPLOT1.YPLOT0)
0018          IF(T.LT.(XM-DELTIM)) GOTO 1
0020          RETURN
0021          END
```